

Saimaan ammattikorkeakoulu
Liiketalous Lappeenranta
Liiketalouden koulutusohjelma
Laskentatoimi

Taavi Aalto

Mikroyritykselle soveltuvan selainkäyttöisen kirjanpito-ohjelman suunnittelu ja toteutus

Opinnäytetyö 2017

Tiivistelmä

Taavi Aalto

Mikroyritykselle soveltuvan selainkäyttöisen kirjanpito-ohjelman suunnittelu ja toteutus, 55 sivua, 2 liitettä

Saimaan ammattikorkeakoulu

Liiketalous Lappeenranta

Liiketalouden koulutusohjelma

Laskentatoimen suuntautumisvaihtoehto

Opinnäytetyö 2017

Ohjaajat: lehtori Marianne Viinikainen ja lehtori Teija Launiainen, Saimaan ammattikorkeakoulu

Tämä tutkimus oli kaksiosainen. Aluksi selvitettiin, mitä vaatimuksia ja rajoitteita kirjanpitolainsäädäntö ja hyvä kirjanpitolain asettavat mikroyrityksen kirjanpito-ohjelmalle. Sen jälkeen ohjelmistotekniikan menetelmiä hyödyntäen suunniteltiin ja toteutettiin kirjanpito-ohjelma, joka täytti ensimmäisen vaiheen vaatimukset.

Tutkimus oli metodiltaan suunnittelutieteellinen. Vaatimusmäärittely perustui hyvään kirjanpitotapaan ja kirjanpitolain määräyksiin juoksevan kirjanpidon ja tilinpäätöksen toteuttamisesta sekä valtioneuvoston asetukseen pien- ja mikroyrityksen tilinpäätöksessä esitettävistä tiedoista. Työn tekijä asetti omia lisävaatimuksia. Vaatimusmäärittelystä edettiin tietokantasuunnittelun sekä ohjelman toteutuksen ja testauksen kautta valmiiseen tuotteeseen. Ohjelmasta oli olemassa koko ajan prototyyppi ja ohjelman kehityksen vaiheet limittyivät päällekkäin. Ohjelma toteutettiin palvelimella suoritettavalla PHP-kielellä ja sen PDO-tietokantarakajapintaa hyödyntäen. Käytettävä tietokanta oli MariaDB. Asiakaspään ohjelmointikielet olivat JavaScript ja HTML.

Työn tuloksena kehitettiin verkkoselaimella käytettävä kirjanpito-ohjelma. Merkittävin suunnittelutieteellinen tulos oli havainto, että pienessäkin projektissa suunnittelu pitää tehdä tarpeeksi kattavasti ennen liian pitkälle edennyttä toteutusta. Tällöin on mahdollista iteroivan palautteen avulla muuttaa suunnittelua tai jopa vaatimusmäärittelyä ja päätyä parempaan lopputulokseen. Ohjelman jatkokehitykselle on hyvät edellytykset. Havaittujen puutteiden korjaamisen jälkeen ohjelma voidaan julkaista avoimen lisenssin alla. Ohjelmaa voi toistaiseksi testata osoitteessa <http://www.palvelen.fi/kipi/>.

Asiasanat: kirjanpito, kirjanpito-ohjelma, tietojärjestelmätiede

Abstract

Taavi Aalto

Requirement Specifications and Implementation of Microbusiness Suitable,
Web-based Accounting Software, 55 pages, 2 appendices

Saimaa University of Applied Sciences

Business Administration Lappeenranta

Degree Programme in Business Administration

Specialisation in Accounting

Bachelor's Thesis 2017

Instructor(s): Ms Marianne Viinikainen and Ms Teija Launiainen, Senior
Lecturers, Saimaa University of Applied Sciences

This study had two phases. First it was examined what kind of requirements and restrictions the law and a good accounting practice impose for an accounting software. After that, an accounting software was designed and constructed by using the methods of software engineering to fulfil the requirements of the first phase.

Design science was used as the research methodology. The requirements were mainly based on the Accounting Act and the Government Decree on the information presented in the financial statements of a small undertaking and micro-undertaking. The author also added some extra requirements for the software. After the requirement analysis, a database should have been modeled before the program was implemented and tested. However, the phases were overlapping and there was a prototype that was improved throughout the process. PHP and its PDO database extension were used as the server side language. MariaDB was used as the database. On the client side JavaScript and HTML were used.

As a result of this thesis a web-based accounting software was implemented. The most remarkable design science result of this thesis was the observation that the design phase should be finished before the program is implemented too far. An early feedback makes it possible to change the design and even the requirements early enough. The program can be improved in the future. After that it can be published under an open source license. The program can be tested at <http://www.palvelen.fi/kipi>.

Keywords: bookkeeping , accounting software, software engineering

Sisällys

1	Johdanto	5
1.1	Tutkimusmenetelmät	6
1.2	Tutkimuksen tavoitteet ja rakenne sekä aiheen rajaus	8
2	Kirjanpidon sääntely ja hyvä kirjanpito tapa	10
2.1	Liiketapahtumien kirjaaminen ja kirjanpitoaineisto	11
2.2	Tilinpäätös	14
2.3	Muuta huomioitavaa	17
3	Kirjanpidon järjestelmä	19
3.1	Kahdenkertainen kirjanpito	19
3.2	Tase ja tuloslaskelma	20
3.3	Yhdenkertainen kirjanpito	21
4	Tietojärjestelmätieteen ja ohjelmistotuotannon teoriaa	21
4.1	Ohjelmistoprojektin vaiheet	21
4.1.1	Vaatusmäärittely	22
4.1.2	Suunnittelu	23
4.1.3	Toteutus ja testaus	26
4.2	Ohjelmistotuotannon mallit	27
5	Case kirjanpito-ohjelman suunnittelu ja toteutus	28
5.1	Ohjelman vaatimusmäärittely	29
5.2	Ratkaisuperiaatteen ja ohjelmistoarkkitehtuurin suunnittelu	32
5.3	Tietokannan suunnittelu	33
5.3.1	Pääkirjanpidon taulut	36
5.3.2	Kirjanpito tilien taulu (tilikartta)	38
5.3.3	Tilinpäätöksen taulut	40
5.3.4	Muut taulut	43
5.4	Ohjelman toteutus ja testaus	44
5.4.1	Suunnitellusta poikkeava toteutus	45
5.4.2	Muu toteutus ja testaus	46
6	Arvio työstä ja jatkokehitys	50
	Kuvat	53
	Taulukot	53
	Lähteet	54
	Liitteet	
	Liite 1 Käyttötapaukset	
	Liite 2 Kirjanpito-ohjelman luokkakaavio	

1 Johdanto

Tässä työssä suunnitellaan ja toteutetaan internet- eli verkkoselaimella (jatkossa myös selain) käytettävä kirjanpito-ohjelma, joka soveltuu mikroyritykselle. Ohjelman potentiaalisia käyttäjiä olisi siten suurin osa kirjanpitovelvollisista. Tämä työ tehdään itselle. Kohdeyrityksenä on erityisesti palveluja myyvä pieni yritys, esimerkiksi konsulttiyritys, jonka ostolaskutuksen määrä on pieni ja ostoreskontran merkitys siis vähäinen.

Mikroyritys on kirjanpitolain 1 luvun 4b §:n mukainen yritys. Mikroyrityksiä ovat ne yritykset, joiden tilinpäätöksessä ja edeltävässä vertailuvuoden tilinpäätöksessä on ylittynyt enintään yksi seuraavista rajoista: tase 350 000 euroa, liikevaihto 700 000 euroa tai tilikauden aikana palveluksessa olleiden henkilöiden lukumäärä keskimäärin 10 henkilöä. Vuoden 2016 alusta voimaan astuneen kirjanpitolain muutoksen takia ammatinharjoittajan ja liikkeenharjoittajan, jonka tilikausi on kalenterivuosi eivätkä tilintarkastuksen rajat ylity, ei tarvitse enää kirjanpitolain 1 luvun 1a §:n mukaan tehdä tilinpäätöstä. (Kirjanpitolaki 30.12.1997/1336.) Muidenkin kuin mikroyritysten tilinpäätöksen laatimista helpotettiin erityisesti liitetietojen osalta. Samalla tase- ja tuloslaskelmakaavoja muutettiin. Nämä muutokset vaativat tekemään muutoksia olemassa oleviin kirjanpito-ohjelmiin.

Kirjanpito-ohjelma on tässä tietokoneohjelma, jota käytetään kirjanpidon tekemisessä apuna. Määritelmä on varsin laava. Sen alle mahtuvat erilaiset ohjelmat taulukkolaskentaohjelmista toiminnanohjausjärjestelmiin. Kirjanpito-ohjelma voi automatisoida kirjanpidon hyvinkin pitkälle, niin että itse kirjauksia ei tarvitse tehdä lainkaan manuaalisesti, vaan ne tehdään automaattisesti. Toisaalta taulukkolaskentaohjelmaan kirjaukset tehdään usein manuaalisesti. Taseen ja tuloslaskelman sekä pää- ja päiväkirjan tuottaminen on helppo automatisoida. Liitetiedot vaativat enemmän ihmisen harkintaa kuten myös tase-erittelyt ja liitetietojen erittelyt. Toimintakertomus hyödyntää vähiten kirjanpitoaineistoa ja vaatii eniten ihmisosaamista. Sitä ei välttämättä ole tarpeellista integroida kirjanpito-ohjelmaan kuin yhtenäisen ulkonäön vuoksi. Yleensäkin integroinnista aiheutuva vaiva suhteessa siitä saatuun hyötyyn asettaa rajat erilaisten ominaisuuksien toteuttamiselle. Ei ole siis yksiselitteisiä vaatimuksia sille, mitä kirjanpito-ohjelman pitää pystyä tekemään.

Osa kirjanpito-ohjelmista on maksullisia ja osa maksuttomia. Toiset ovat omalle koneelle asennettavia ja toiset selainkäyttöisiä. Osassa on suomenkielinen käyttöliittymä. Suomalainen Devlab Oy ylläpiti 1.2.2017 asti tukkureille ja kaupoille tarkoitettua selainkäyttöistä GPL-lisensoitua Pupesoft-toiminnanohjausjärjestelmää, joka on nykyisin myyty Sprintit Oy:lle (Devlab 2017a). Pupesoft sisältää muun ominaisuuden lisäksi myös kirjanpidon (Devlab 2017b). Odoo on selain-pohjainen kirjanpitomoduulin sisältävä toiminnanohjausjärjestelmä (Odoo s.a. 2017). Tilitin on suomenkielinen Java-pohjainen ilmainen kirjanpito-ohjelma (Helineva 2011, 31). Kirjanpito-ohjelmista puuttuu suomenkielinen selainpohjainen versio, joka on tarkoitettu vain kirjanpitoa varten ja jonka voisi myös asentaa omalle koneelle.

Kirjanpito-ohjelmia on paljon erilaisia. Niiden tekemisestä on kuitenkin julkaistu vain vähän tutkimuksia, vaikka ohjelmistotekniikkaa eli ohjelmistojen valmistamiseen liittyvää tiedettä on tutkittu paljon. Tommi Helineva on tehnyt Metropolia ammattikorkeakoulussa opinnäytetyön kirjanpito-ohjelman toteuttamisesta. Sen tuloksena syntyi Tilitin-kirjanpito-ohjelma (Helineva 2011, 32). Sharma ja Sarangdevot tutkivat aspektiperusteisen ohjelmoinnin (AOP) soveltuvuutta kirjanpito-ohjelman kehittämiseen ja erityisesti muutosten tekemiseen ohjelmaan (Sharma & Sarangdevot 2010). On mahdollista, että avointen lähdekoodien ohjelmista ja niiden kehittäjäyhteisöistä on selvitettävissä, miten ohjelma on tehty. Esimerkiksi Ari Marin on tehnyt opinnäytetyössään Tampereen ammattikorkeakouluun muutoksia Pupesoft-toiminnanohjausjärjestelmän sähköisten laskujen ohjelmakoodiin, vaikka ei suoraan kirjanpidon osaan muutoin kuin tiliöinnin osalta (Marin 2014, 40-42).

1.1 Tutkimusmenetelmät

Ammattikorkeakouluissa tehdään paljon toiminnallista tutkimusta (practice-based research), jonka tuloksena syntyy toimeksiantajalle tehtävä tuote eli produkti. Se voi olla myös tapahtuma, opastus tai ohjeistus. Toiminnallinen opinnäytetyö koostuu kahdesta osasta: toiminnallisesta osasta, jonka tuloksena syntyy tuote, ja toiminnallisen osan selostuksesta ja arvioinnista, jonka tuloksena syntyy opinnäytetyöraportti. Opinnäytetyön kirjoittamista varten tuotteen tekemisprosessi on syytä dokumentoida. (Vilkka & Airaksinen 2003, 19, 38 ja 65.)

Toiminnallisen opinnäytetyön suunnittelussa tulee vastata kysymyksiin, mitä, miten ja miksi tehdään. Toiminnallisessa opinnäytetyössä ei esitetä tutkimusongelmaa ja tutkimuskysymyksiä. (Vilka & Airaksinen 2003, 26 ja 30.) Vilkan mukaan ne voidaan kuitenkin esittää, jolloin kyse on enemmänkin opinnäytetyön lähtökohtien esittämisestä. Toiminnallinen opinnäytetyö on toiminnallista kehittämistä, jossa tutkimuksen työkaluja käytetään viitekehyksenä apuna vain tuotoksen tekemisessä. (Vilka 2016.)

Tutkimuksellisessa kehittämisessä katsontasuunta on kehittämistoiminnasta tutkimukseen päin painon ollessa kehittämistoiminnassa ja käytännön ongelmista nousevien kysymysten selvittämisessä. Kehittävässä tutkimuksessa taas katsontasuunta on tutkimuksesta kehittämiseen päin. Tietoa tuotetaan tiedeyhteisön sääntöjen mukaan lähtien liikkeelle tutkimuskysymyksestä edeten kehittämistoiminnan suuntaan. (Toikko & Rantanen 2009, 21–22.) Seuraavassa kappaleessa kuvattava suunnittelutieteellinen tutkimus on tämän työn tekijän mielestä kehittävä tutkimusta, toiminnallinen tutkimus taas tutkimuksellista kehittämistä.

Järvisen ja Järvisen mukaan suunnittelututkimus (design research) eli suunnittelutiede (design science) vastaa muun muassa kysymyksiin *millainen pitäisi tietyn innovaation olla ja kuinka meidän tulisi se rakentaa?* Van Akenin (2004) mukaan suunnittelutieteen tulos ei ole toteutettu artefakti eli innovaatio vaan sen suunnittelun tuottama suunnittelutietämys, jota voidaan käyttää jatkossa hyväksi muissa suunnittelu- ja konstruointiongelmassa. Järvisen käsitys Van Akenin suunnittelutietämyksen jakautumisesta kolmeen osaan on lopputuloksen (artefaktin) suunnittelu ja määrittely, lopputuloksen saavuttamiseen tarvittavan prosessin suunnittelu (tekniset, sosiaaliset ja tietoresurssit) ja toteutuksen eli käytännön toimenpiteiden suunnittelu. Tutkimustulokset ovat usein heuristisia preskriptioita, jotka kuvaavat, miten kannattaa toimia tai mitä pitää tehdä, jotta päästäisiin haluttuun tilaan. Marchin ja Smithin (1995) mukaan suunnitteluongelma osoittautuu ratkaisuksi, mikäli artefakti onnistutaan toteuttamaan. (Järvinen & Järvinen 2011, 103–105.)

Myös toimintatutkimus (action research) kävisi lähestymistavaksi. livari ja Venable pitävät sitä tutkimusmetodin ja suunnittelutiedettä taas useamman tutkimusmetodin käytön sallimana tutkimussuuntauksena (livari & Venable 2009). Tässä

työssä käytetään kuitenkin lähinnä suunnittelutieteellistä lähestymistapaa. Kuten havaitaan, ovat edellä mainitut tutkimusmenetelmät hyvin samankaltaisia ja niistä muutkin soveltuisivat hyvin tähän työhön. Yhteistä niille on pyrkimys vastata kysymykseen, millainen reaali maailman pitäisi olla eikä millainen se on, kuten käsitteelliset-teoreettiset ja empiiriset tutkimukset tekevät (Järvinen & Järvinen 2011, 9).

Tässä opinnäytetyössä painotetaan Van Akenin mukaista suunnittelun osuutta, vaikka suunnittelua ei jaeta yhtä hienopiirteisesti. Myös Marchin ja Smithin painottama artefaktin toteuttaminen on työn tärkeä osa.

1.2 Tutkimuksen tavoitteet ja rakenne sekä aiheen rajaus

Tässä luvussa esitetään tutkimuksen tavoite, rakenne ja aiheen rajaus. Lopussa esitetään tutkimuskysymykset. Työssä kehitetään mikroyritykselle soveltuva verkkoselaimella käytettävä kirjanpito-ohjelma. Kehittäminen on kolmivaiheinen. Ensin tutkitaan ja määritellään, millainen kirjanpito-ohjelman pitää olla. Sitten suunnitellaan kirjanpito-ohjelma ja lopuksi toteutetaan se. Lain ja muun sääntelyn lisäksi ohjelmistotekniikan menetelmät ja kirjanpidon järjestelmä muodostavat työn viitekehyksen.

Alussa tehtävässä vaatimusmäärittelyssä (requirement specification) kuvataan järjestelmälle asetetut vaatimukset – niin lainsäädännölliset kuin toiminnalliset. Ohjelman tulee noudattaa kirjanpitolakia ja valtioneuvoston asetusta pien- ja mikroyrityksen tilinpäätöksessä esitettävistä tiedoista sekä mahdollistaa kirjanpidon tekemisen hyvän kirjanpitotavan mukaan.

Kun vaatimukset on määritetty ja ohjelma kuvattu (spesifikaatio), aloitetaan ohjelman suunnittelu. Tavoitteena on esittää tarvittavat menetelmät, joilla määrittelyistä ja ohjelman kuvauksesta päädytään valmiiseen tuotteeseen. Tähän lopputuloksen saavuttamisen suunnitteluprosessiin yhdistetään käytännön toteutuksen pohdinta, vaikka ne edellä esitettiin suunnittelutietämyksen eri osina.

Koska opinnäytetyön tekemiseen varattu aika ei riitä niin hyvän kirjanpito-ohjelman tekemiseen kuin olisi suotavaa, joudutaan tekemään priorisointia. Ohjelmasta luodaan ensin suppea toimiva prototyyppi, jota lähdetään kehittämään.

Toiminnallisuus asetetaan käytettävyyden edelle. Tärkeää on, että jokin ominaisuus saadaan ensin toimimaan. Sitten ominaisuuden pitää toimia oikein ja lainmukaisesti.

Kun nämä tavoitteet on saavutettu, kiinnitetään huomiota mm. tietoturvaan, käytettävyyteen, raportoinnin muotoon ja selkeyteen, laajennettavuuteen, integraatioon muihin järjestelmiin ja tehokkuuteen sikäli kuin aikaa jää. Suurin osa tästä osuudesta jäänee toteuttamatta ja sitä käsitellään vain suunnittelun yhteydessä. Tosin jo suunnitteluvaiheesta alkaen pidetään vähemmän painotetut tavoitteet mielessä ja pyritään välttämään valintoja, joita pitää jatkossa muuttaa. Mitä myöhemmin virhe tai muutostarve yleensä nimittäin havaitaan, sitä enemmän työtä korjaaminen tai muuttaminen vaatii.

Toteutus ja suunnittelu käsittävät vain kirjanpito-ohjelman. Esimerkiksi reskontrat ja laskutus rajataan työn ulkopuolelle. Samoin tehdään muun muassa veroilmoituksen, varastokirjanpidon ja käyttöomaisuuskirjanpidon kanssa. Johdon laskentatoimen tarpeita ei huomioida lainkaan eikä siten esimerkiksi kustannuspaikkoja toteuteta. Edellä mainittuja saatetaan käsitellä suunnittelun osalta, mikäli ne jontekin rajaisivat ohjelman jatkokehitystä poissulkevasti.

Tutkimusongelma on suunnittelutieteellinen – tietyt vaatimukset täyttävän kirjanpito-ohjelman suunnittelu ja valmistus. Suunnittelutieteellistä tietoa on se, millä keinoin saadaan valmistettua halutunlainen kirjanpito-ohjelma ja mitä tekemisprosessissa opitaan.

Tutkimuskysymys on kaksiosainen ja se voidaan edellä kuvatut rajoitteet huomioiden esittää yleisesti:

1. Millaisia vaatimuksia ja rajoitteita lainsäädäntö ja hyvä kirjanpito-asettavat mikroyritykselle soveltuvalla kirjanpito-ohjelmalle?
2. Miten vaatimukset ja rajoitteet täyttävä kirjanpito-ohjelma suunnitellaan selainpohjaisena ja miten se toteutetaan?

Ensimmäiseen tutkimuskysymykseen vastataan luvussa 2, jossa käydään lävitse kirjanpidon lainsäädäntöä ja säätelyä, sekä luvussa 3, jossa esitetään kirjanpidon järjestelmä. Toisen tutkimuskysymykseen vastataan luvussa 5, jossa kerrotaan,

miten tämän opinnäytetyön tuloksena syntynyt kirjanpito-ohjelma suunniteltiin ja toteutettiin käytännössä. Luku 4 käsittelee ohjelmiston tai tietojärjestelmän suunnittelun ja toteutuksen teoreettista pohjaa yleisellä tasolla.

2 Kirjanpidon sääntely ja hyvä kirjanpitolaki

Tässä luvussa vastataan ensimmäiseen tutkimuskysymykseen käsittelemällä kirjanpitolainsäädännöstä niitä asioita, jotka vaikuttavat kirjanpito-ohjelman suunnitteluun ja toteutukseen. Käsiteltävien asioiden yhteydessä otetaan kantaa siihen, mitkä asiat toteutetaan juuri tämän opinnäytetyön kirjanpito-ohjelmassa. Luvussa 2.1 käydään lävitse juoksevaan kirjanpitoon liittyviä asioita ja luvussa 2.2 tilinpäätökseen liittyviä asioita.

Kirjanpitolaki on yleislaki, joka väistyy erityislainsäädännön tieltä. Kirjanpitoa säätelevää erityislainsäädäntöä on erityisesti yhtiömuotoja säätelevässä lainsäädännössä kuten esimerkiksi osakeyhtiö-, osuuskunta-, säätiö- ja asunto-osakeyhtiölaissa. Erillislakien säädökset säätelevät tyypillisesti tilinpäätöstä ja siellä erityisesti liitetietojen antamiseen liittyviä asioita. Lisäksi verolainsäädännöllä on kiinteä yhteys kirjanpitoon.

Lakien lisäksi mikroyrityksen kirjanpitoa säätelevät valtioneuvoston asetus pien- ja mikroyrityksen tilinpäätöksessä esitettävistä tiedoista ja kirjanpitolautakunnan (KILA) yleisohjeet ja lausunnot, jotka tulkitsevat hyvää kirjanpitolakia. Merkityksellisin KILA:n yleisohjeista työn kannalta on kirjanpitolautakunnan yleisohje kirjanpidon menetelmistä ja aineistoista 1.2.2011. Se on kuitenkin annettu ennen kirjanpitolain muutosta vuoden 2016 alusta alkaen, joten sitä ei voida kaikilta osin soveltaa. Ohjeessa on viittauksia kauppa- ja teollisuusministeriön vuodelta 1998 annettuun päätökseen kirjanpidossa käytettävistä menetelmistä. Nykyisin kirjanpitolain soveltamisesta antaa lain mukaan ohjeita ja lausuntoja vain työ- ja elinkeinoministeriön asettama kirjanpitolautakunta (Kirjanpitolaki 30.12.1997/1336). Kirjanpitolain on poistettu maininta vastaavan ministeriön suorista päätöksistä.

Merkittävä osa kirjanpitolainsäädännöstä käsittelee kirjausten tekemistä ja sellaisia asioita, jotka eivät vaikuta kirjanpito-ohjelmaan, mikäli ohjelmaa ei automatisoida kovin pitkälle. Jos ohjelmassa olisi esimerkiksi tekoälyä ja sen odotettaisiin

kykenevän muodostamaan kirjanpito tilinpäätökseen asti lähes itsenäisesti, olisi suurin osa lainsäädännöstä käsiteltävä tässä työssä. Näin ei kuitenkaan ole, vaan kyse on yksinkertaisesta kirjanpito-ohjelmasta.

2.1 Liiketapahtumien kirjaaminen ja kirjanpitoaineisto

Kirjanpitolain 2 luku säätelee juoksevaa kirjanpitoa. Luvun 2 §:n mukaan *Liiketa-*
pahtumat merkitään asian mukaan kirjanpitotileille (kirjaus). Saman pykälän mu-
kaan *Kirjanpitovelvollisella tulee olla kultakin tilikaudelta kirjanpitotileistä selkeä*
ja riittävästi eritelty luettelo, joka selittää tilien sisällön (tililuettelo). Kirjauksen on
5 §:n mukaan *perustuttava päivättyyn ja järjestelmällisesti numeroituun tai vas-*
taavalla tavalla yksilöityyn tositteeseen, joka todentaa liiketapahtuman. Liiketapaht-
tuman, tositteen ja kirjauksen välisen yhteyden tulee olla vaikeuksitta todetta-
vissa. Myös liitetietotositteiden ja liitetietojen välisen yhteyden on oltava 5 b §:n
mukaan vaikeuksitta todennettavissa. Lisäksi hallituksen esityksen mukaan *tosit-*
teesta tulee käydä selville momentissa mainitun päiväyksen ja numeroinnin
ohella tositteen antajan nimi, liiketapahtuman sisältö ja rahamäärä. Numeroinnin
vaatimusta on lavennettu muullakin tavalla yksilöityyn tositteeseen siten, että ky-
seeseen tulee esimerkiksi tietokannan hakuavain tai looginen osoitelinkki. Vaati-
mus tiliöintimerkinnän tekemisestä tositteeseen päätettiin korvata vaatimuksella
että *tositteen, liiketapahtuman ja kirjauksen välisen yhteyden pitää olla vaikeuk-*
sitta todettavissa. (Kirjanpitolaki 30.12.1997/1336; HE 89/2015.)

Lain mukaan kirjanpito-ohjelmassa pitäisi siis olla vähintään kirjanpitotilejä ja
niille riittävän selkeät nimet (selitykset). Laki ei kuitenkaan vaadi tilien nimien tal-
lentamista kirjauksessa. Pelkkä tilinumero riittää. Päiväys ja tositenumero tai muu
tositteen yksilöivä tunnus liitetään kirjaukseen epäsuorasti tositteen kautta. Tosi-
tenumero ei siis ole kirjauksessa välttämätön, jos käytetään muuta tositteen yk-
silöivää tunnusta. Vaatimus tositteen ja kirjauksen välisen yhteyden todennetta-
vuudesta vaikeuksitta kuitenkin vaatinee tositenumeroiden käyttöä kirjauksissa
ainakin siinä tapauksessa, että tositteet eivät ole sähköisessä muodossa ja oh-
jelman käytettävissä. Tässä työssä toteutettavan kirjanpito-ohjelman kirjauksissa
vaaditaan kirjanpitotilin numero ja sekä tositenumero että päiväys. Työn kirjan-
pito-ohjelmassa toteutetaan myös lain vaatima tililuettelon tulostaminen.

Kirjanpitolain 2 luku 4 § edellyttää, että *kirjauksia voidaan tarkastella aikajärjestyksessä ja asiajärjestyksessä* (Kirjanpilolaki 30.12.1997/1336). Aikaisemmin laissa puhuttiin tarkastelun sijasta kirjaamisesta aika- ja asiajärjestykseen. Sana- muotoa muutettiin vastaamaan nykyisiä tietojärjestelmiä ja käytäntöjä. (HE 89/2015.) Liiketapahtumien luettelo aikajärjestyksessä muodostaa päiväkirjan ja luettelo asiajärjestyksessä eli tileittäin muodostaa pääkirjan (Tomperi 2016, 16). Paperinen tilinpäätös edellyttäne yhä päivä- ja pääkirjan muodostamista, mutta sähköinen kirjanpito sallinee esittää vain tarpeelliset tilit eikä koko pääkirjaa, kun- han kirjausketju säilyy. Tämä pykälä vahvistaa vaatimusta päiväyksen lisäämi- sestä kirjanpito-ohjelman kirjaukseen, jotta kirjaukset voidaan järjestää aikajär- jestykseen. Tämän työn kirjanpito-ohjelmassa toteutetaan päivä- ja pääkirjan tar- kastelumahdollisuus, mutta päivä ja pääkirjoja ei erikseen toteuteta tietojärjestel- män tasolla.

Tilinpäätös johdetaan kirjanpitolain 2 luvun 2 § 3 momentin mukaan pääkirjanpi- dosta, josta on 6 §:n mukaan oltava kirjausketju (audit-trail) sekä tositteisiin että tilinpäätökseen. Kirjausketjun liiketapahtumista ja tositteista tilinpäätökseen pää- kirjanpidon kautta on oltava katkeamaton molempiin suuntiin. KILA:n menetelmä- ohjeen mukaan kirjausten yhteys pääkirjanpidosta tuloslaskelmaan ja taseeseen sekä toiseen suuntaan voidaan toteuttaa esimerkiksi tilikohtaisella taseella ja tu- loslaskelmalla. Siinä tase ja tuloslaskelma on laadittu pääkirjan tileittäin eritel- tynä. Koneellisella tietovälineellä voidaan käyttää linkkiä pääkirjanpidon tililtä vas- taavalle tilinpäätöstilille ja toiseen suuntaan tilinpäätöstililtä niihin pääkirjan tilei- hin, jotka päätetään kyseiselle tilinpäätöstilille. Tietojärjestelmän kohdalla voi- daan hakea tarvittavat tiedot näytölle. Kirjausketju tositteesta pääkirjanpitoon ja toiseen suuntaan voidaan toteuttaa tositenumeron tai muun tositteen yksilöivän tiedon perusteella, joka on tallennettu pääkirjanpitoon. Koneellisella tietoväli- neellä olevasta pää- tai päiväkirjasta voi myös olla linkki tositteeseen ja toisin päin. Tietojärjestelmä voi myös yhdistää tiedot tositumerkintöjen perusteella. (Kir- janpilolaki 30.12.1997/1336; Kirjanpilolautakunta 2011, 16-17.)

Tässä työssä toteutettavassa kirjanpito-ohjelmassa yhteys pääkirjanpidon ja tilin- päätöksen välillä toteutetaan tilikohtaisella taseella ja tuloslaskelmalla, sillä ne voi tulostaa tai säilyttää sähköisesti tietojärjestelmästä riippumattomana. Kirjausketju

tositteesta pääkirjanpitoon toteutetaan niin, että edellytetään tositteiden numeerointia. Silloin on helpompi toteuttaa kirjanpito-ohjelma, jolla kirjausketju säilyy tietojärjestelmästä riippumattomana, paperilla tai tiedostoissa. Toiseen suuntaan pääkirjanpidosta tositteeseen päin toteutetaan myös linkki, koska se on teknisesti helppo tehdä. Jos kirjanpito-ohjelmassa käytettäisiin konenäköä ja hahmontunnistusta skannattujen tositteiden käsittelyssä tai käytössä olisi sähköinen laskutus, voisi tositemerkinnät kirjata automaattisesti. Tietojärjestelmä voisi sitten yhdistää tositteet, mutta se olisi tässä työssä liian vaativaa saavutettuun hyötyyn nähden.

Kirjanpitolain 2 luvun 7 § mukaan kirjanpitoa *ei saa muuttaa tai poistaa tilinpäätöksen laatimisen jälkeen* eikä sen jälkeen, kun kirjanpidosta on annettu ilmoitus viranomaiselle (Kirjanpitolaki 30.12.1997/1336). Aikaisemmin laissa sanottiin, että kirjanpitomerkintää ei saa poistaa eikä tehdä epäselväksi. Laissa ei otettu eikä oteta nytkään kantaa siihen, onko kirjanpito tehty kynällä vai tietokoneella. Käytännössä kirjanpito-ohjelmissa on jo pitkään ollut mahdollisuus korjata tehtyjä virheitä. Taulukkolaskennassa tehdyn kirjanpidon korjaaminen on hyvin helppoa, mutta se voi olla lain vastaista. Jos muuttamisen lainmukaisuus jätetään käyttäjän huoleksi, on kirjanpito-ohjelman toteuttaminen huomattavasti helpompaa. Halutessaan kirjanpitäjä pystyy nimittäin aina väärentämään kirjanpidon, vaikka sitten viimekädessä tekemällä sen uudestaan alusta asti erilaisena. Tosin pilvipohjaisessa kirjanpidossa käyttäjä ei välttämättä pysty vaikuttamaan ulkopuolisen tahon ylläpitämän kirjanpito-ohjelman kirjausten yhteydessä tehtyihin aikaleimoihin. Jos kirjanpito kuitenkin tulostetaan paperille, jäisi tässäkin monesti mahdollisuus toimia vastoin lakia. Lähtökohtaisesti tässä työssä kirjanpito-ohjelma suunnitellaan ja toteutetaan siksi oletuksella, että kirjanpitäjä toimii rehellisesti ja lain mukaisesti.

Kirjanpitolain 2 luvun 7 a § mukainen velvollisuus kirjanpitoaineistojen luettelon tekemisestä jätetään kirjanpitovelvollisen tehtäväksi, eikä kirjanpito-ohjelma tässä työssä tuota luetteloja. Samoin kirjanpitoaineiston 2 luvun 9 § mukainen säilyttäminen jätetään kirjanpitovelvollisen tehtäväksi. Pykälän mukaan kirjanpitoaineiston säilyttäminen pitää tehdä niin, että se on selväkielisesti tarkasteltavissa Suomessa (Kirjanpitolaki 30.12.1997/1336). Kantaa ei oteta nykyään enää

säilytyspaikkaan. Tosin muu lainsäädäntö saattaa ottaa kantaa säilytyspaikkaan, mikäli käsitellään henkilötietoja, joita saattaa sisältyä ainakin tositteisiin ja myös kirjausten selitteisiin.

KILA:n menetelmäohjeen mukaan kirjanpitovelvollinen voi kirjanpitoaineiston säilyttämistä varten teknisesti lukita *tietojärjestelmän tietokannan siten, ettei tietoja lukitsemisen jälkeen enää voida muuttaa, lisätä tai poistaa kirjanpitojärjestelmän, tietokannan hallinta-ohjelman tai muun ohjelmiston tarjoamin välinein kirjanpitovelvollisen tai tämän käyttämän ulkoistetun kirjanpito palvelun tuottajan toimesta* (Kirjanpitolautakunta 2011, 30). Tämä on opinnäytetyön tekijän mielestä mahdotonta ja kirjanpitolain muutoksen perusteluissakin päätettiin luopua säilytyksen teknisten yksityiskohtien säätelystä mahdollistaen kirjanpidon toteutuksen *kulloisenkin tilanteen vaatimusten ja tietotekniikan mahdollisuuksien mukaan* (HE 89/2015). Yleensäkin kirjanpidon säilyttäminen pelkästään tietokannassa ei ole kirjanpitovelvollisen kannalta järkevää, sillä viimekädessä vastuu kirjanpidon järjestämisestä ja säilytyksestä on sopimuksista huolimatta kirjanpitovelvollisella. Tässä työssä tehtävässä toteutuksessa oletetaan, että kirjanpitovelvollinen säilyttää kopiot tilinpäätöksestä itsellään, vaikka ohjelmaan toteutetaan tilinpäätösten säilyttäminen. Tähän kirjanpito-ohjelmaan ei integroida sähköistä allekirjoitusta. Kirjanpitovelvollinen voi halutessaan allekirjoittaa tilinpäätöksen erikseen HST-kortilla, jolloin kirjanpitoaineisto on mahdollista säilyttää kokonaan sähköisenä. Jos vain tilinpäätös tulostetaan ja allekirjoitetaan, kirjanpitovelvollinen voi laskea KILA:n ohjeen mukaan sähköisesti säilytettävistä tiedostoista ulkopuolisilla ohjelmilla tarkistesummat tiedostojen muuttumattomuuden varmistamiseksi ja lisätä tiedot vaikkapa allekirjoitettavan tilinpäätöksen liitetietoihin.

2.2 Tilinpäätös

Tilinpäätöksen antamista säätelee kirjanpitolain 3 luku ja valtioneuvoston asetus pien- ja mikroyrityksen tilinpäätöksessä esitettävistä tiedoista (PMA-asetus). Kirjanpitolain 3 luvun 1 § määrittelee aluksi, että kirjanpitovelvollisen tulee esittää edellisen tilikauden vertailutiedot kustakin taseen ja tuloslaskelman erästä siten, että vertailutiedot on tarvittaessa muutettu vastaamaan kuluvan tilikauden eriä. Kirjanpitolain mukaan toimintakertomusta ei tarvitse liittää tilinpäätökseen, kun

kirjanpitovelvollinen on pien- tai mikroyritys. Osa mikroyrityksistä joutuu liittämään myös tilintarkastusta koskevat tiedot tilinpäätökseensä. (Kirjanpitolaki 30.12.1997/1336.) Niissä tapauksissa, joissa kirjanpitovelvollinen joutuu antamaan toimintakertomuksen muun lain nojalla, on se luontevaa tehdä erillisellä tekstinkäsittelyohjelmalla. Toimintakertomusta ei toteuteta tähän kirjanpito-ohjelmaan. Koska tilintarkastuskertomuksen antaa ulkopuolinen tilintarkastaja, ei sitäkään toteuteta tähän ohjelmaan. Vertailutietojen muuttamista vastaamaan kulu- van vuoden lukuja ei toteuteta tässä ohjelmassa, vaan käyttäjän on tehtävä se käsin.

Tilinpäätös on *esitettävä suomen tai ruotsin kielellä euromääräisenä* ja se on päivittävä ja allekirjoitettava. *Tilinpäätöstä varmentamaan laaditaan tase-erittelyt ja liitetietojen erittelyt* kirjanpitolain mukaan. (Kirjanpitolaki 30.12.1997/1336.) Kirjanpitolaki ei nykyään enää käytä termiä tasekirja, vaan koko tilinpäätöksen voi nykyään allekirjoittaa sähköisesti, jolloin kaiken aineiston voi säilyttää sähköisesti. Päiväys ja allekirjoituskohta on helppo toteuttaa liitetietojen loppuun. Sähköisen allekirjoituksen toteuttaminen vaatisi vahvaa sähköistä tunnistamista. Tase-erittelyiden pohjat on helppo tuottaa automaattisesti selitteiden perusteella. Tässä käyttäjälle tarjotaan tase-erittelyihin vaadittavat tiedot ohjelmasta. Käyttäjän tulee kuitenkin itse muokata tase-erittelyt valmiiksi. Myöskään liitetietojen erittelyjä ei toteuteta ohjelmassa.

Mikro- ja pienyrityksen tuloslaskelmakaava esitetään PMA-asetuksen liitteen I kululajikohtaisen kaavan tai asetuksen liitteen II toimintokohtaisen kaavan mukaan. Molemmat yritystyyppit saavat käyttää samaa lyhennettyä tuloslaskelmakaavaa. Mikroyritys saa jättää esittämättä liitteiden kirjaimin merkityt alakohdat. (Valtioneuvoston asetus pien- ja mikroyrityksen tilinpäätöksessä esitettävistä tiedoista 1753/2015.) Tässä työssä toteutettavassa kirjanpito-ohjelmassa käyttäjä ei pääse muokkaamaan taseen ja tuloslaskelman rakennetta. Työssä ei toteuteta lyhennettyä kaavaa, jätetään esittämättä kirjaimin merkityt alakohdat ja toteutetaan vain kululajikohtainen tuloslaskelmakaava.

Mikro- ja pienyritys käyttävät samaa PMA-asetuksen V liitteen tasekaavaa. Molempien vastaavaa puolta voidaan lyhentää siten, että *esitetään erikseen aineet-*

tomat hyödykkeet, aineelliset hyödykkeet ja sijoitukset; sekä vaihtuvissa vastavissa esitetään erikseen vaihto-omaisuus, saamiset, rahoitusarvopaperit sekä rahat ja pankkisaamiset. Mikroyrityksen ei tarvitse eritellä pitkäaikaisia saamia toisin kuin pienyrityksen. Mikroyrityksellä *vastattavaa puolella esitetään erikseen oma pääoma, tilinpäätössiirtojen kertymä, pakolliset varaukset ja vieras pääoma.* Pienyrityksen on eriteltävä oma pääoma tarkemmin ja annettava tieto vieraan pääoman pitkäaikaisten velkojen määrästä kun taas mikroyrityksen tulee antaa tieto vieraan pääoman lyhytaikaisten velkojen määrästä. Mikroyrityksen tulee eritellä omasta pääomasta *ne erät, jotka on säädetty omaan pääomaan kuuluviksi mikroyritystä koskevassa laissa sekä edellisten tilikausien voitto (tappio) ja tilikauden voitto (tappio).* (Valtioneuvoston asetus pien- ja mikroyrityksen tilinpäätöksessä esitettävistä tiedoista 1753/2015.) Tässä työssä toteutetaan mikroyrityksen tasekaava lyhimmillään vain liiketoimintaa harjoittavalle yritykselle.

Asetuksen 2 luvun 6 § mukaan eria saa esittää vaadittua tarkempana, mikäli se ei vaaranna oikean ja riittävän kuvan syntymistä. Sitä itse asiassa vaaditaankin, *jos se on tarpeen tilikauden tuloksen muodostumiseen vaikuttaneiden tekijöiden tai tase-erien selventämiseksi.* Nimikettä ei merkitä tuloslaskelmaan tai taseeseen, jos sen kohdalle ei tule lukua tilikaudelle tai vertailuvuodelle (Valtioneuvoston asetus pien- ja mikroyrityksen tilinpäätöksessä esitettävistä tiedoista 1753/2015.) Kuten edellä todettiin, ei käyttäjälle anneta tässä toteutettavassa kirjanpito-ohjelmassa mahdollisuutta muokata tilinpäätöksen rakennetta, joten tilinpäätöstä ei voi esittää ohjelman esitystä tarkempana. Tällöin ohjelmasta ei voida myöskään tuottaa kiinteistökaavan tai säätiö- ja yhdistyskaavan mukaista tilinpäätöstä ohjelmaa muokkaamatta. Kirjanpito-ohjelmassa pitäisi yleensä olla mahdollisuus muokata tilinpäätöskaavoja. Käyttäjälle annetaan tässä kuitenkin mahdollisuus lisätä omia kirjanpitotilejä, mutta ei tilinpäätöstilejä. Käyttäjä voi pääkirjan perusteella muokata tilinpäätöstä tarkemmaksi ohjelman ulkopuolella tai toteuttaa kustannuspaikkaseurantaa omien kirjanpitotilien avulla.

PMA-asetuksen 3 luku luettelee ilmoitettavat liitetiedot. Niiden osalta pien- ja mikroyritys noudattavat pienyrityksen liitetietojen antovaatimusta. Poikkeuksena on 3 luvun 5 §:n mukainen sijoituskiinteistöjen ja rahoitusvälineiden merkitseminen

käypään arvoon. Se on kirjanpitolain 5 luvun 2a ja 2b §:n mukaan kielletty mikroyritykseltä. Asetuksen 3 luvun 13 §:n mukaiset toimintakertomusta vastaavat tiedot eivät enää ole pakollisia mikroyritykselle. Muut pykälät koskevat tilinpäätöksen laatimisperiaatteita, poikkeuksellisia eriä, tilikauden jälkeisiä olennaisia tapahtumia, arvonkorostusrahastoja, pitkäaikaisia lainoja, annettuja vakuuksia ja taseen ulkopuolisia sitoumuksia ja järjestelyjä sekä eläkevastuita, tietoja emoyrityksestä, lainoja toimitusjohtajalle ja hallintoelimiin kuuluville henkilöille sekä heidän puolestaan annettuja sitoumuksia, liiketoimia intressitahojen kanssa, henkilöstöä, omien osakkeiden ja osuuksien hankintoja ja luovutuksia. (Kirjanpitolaki 30.12.1997/1336; Valtioneuvoston asetus pien- ja mikroyrityksen tilinpäätöksessä esitettävistä tiedoista 1753/2015.) Pykälä 5 ja 13 ei toteuteta tässä työssä. Sen sijaan liitetiedot kysellään 3 luvun muista pykälistä sen mukaan kuin niistä tarkemmin säädetään asetuksessa.

2.3 Muuta huomioitavaa

Kirjanpitovelvollinen on rekisterin pitäjä, kun hän tallentaa kirjanpito-ohjelmaan henkilötietoja, esimerkiksi työntekijän nimen. Koska kirjanpito-ohjelmaan mitä ilmeisimmin tallennetaan selitteenä työntekijän nimi, muodostunee siitä henkilörekisteri. Henkilötietolain 5 § asettaa rekisterinpitäjän lisäksi sille, joka itsenäisenä elinkeinon- tai toiminnanharjoittajana toimii rekisterinpitäjän lukuun huolellisuusvelvoitteen. Se pitää sisällään muun muassa velvoitteen *noudattaa huolellisuutta ja hyvää tietojenkäsittelytapaa*. Henkilötietolain 32 § asettaa samanlaisen tietojensuojaamisvelvoitteen muun muassa *toteuttaa tarpeelliset tekniset ja organisatoriset toimenpiteet henkilötietojen suojaamiseksi asiattomalta pääsylvä tietoihin*. (Henkilötietolaki 22.4.1999/523.) Kuten edellä todettiin, tietojen syöttäminen jätetään kirjanpitäjän vastuulle. Tästä mainitaan ohjelman ohjeessa.

Kirjanpitolaki ei tarkkaan ottaen eksplisiittisesti edellytä rahamääräisten summien kirjaamista. Rahojen kirjaamisista puhutaan vain käteissuoritusten yhteydessä ja rahamäärien merkitsemisestä tositteiden yhteydessä. Koska kirjauksissa kuitenkin kirjataan nimenomaan rahamääriä eri tileille, kuten luvusta 3 käy ilmi, asetetaan rahamäärän kirjaaminen tässä kirjanpitolainsäädännön implisiittiseksi vaatimukseksi. Koska tätä ohjelmaa ei myöskään suunnitella ymmärtämään tosittei-

den sisältöjä, pitää tositteiden rahasummat siitakin syystä kirjata tileille, jotta voidaan muodostaa pääkirjanpito ja siitä tilinpäätös. Käytännössä kirjaukseen kannattaa liittää aina kirjanpito-ohjelmissa rahamäärä, sillä se nopeuttaa huomattavasti ohjelman toimintaa. Jos rahamääräiset arvot haettaisiin aina niitä tarvittaessa tositteilta, pitäisi tositteet säilyttää yhtä pitkään kuin kirjanpito-ohjelmatkin.

Kirjanpitolaki ei vaadi selitteiden käyttöä. Tässä ohjelmassa, kuten kirjanpito-ohjelmissa yleensä, käytetään tosite- ja kirjauskohtaisia seliteitä, sillä ne selkeyttävät ohjelman käyttöä ja kirjanpidon lukemista. Lisäksi niitä voidaan hyödyntää esimerkiksi tase-erittelyissä.

PMA-asetuksen tase- ja tuloslaskelmakaavoissa ei esitetä lainkaan lukuja. KILA:n yleisohjeessa tuloslaskelman ja taseen esittämisestä vuodelta 2006 annetaan mahdollisuus jättää etusarakkeet pois, jolloin tulee merkitä tietyt välisummat. Koska lainsäädäntö on muuttunut tämän jälkeen, on luontevaa ajatella, että näytetään kaikkien lain vaatimien nimikkeiden (väli)summat nimikkeiden perässä. KILA:n tasemallissa vastaavan ja vastattavan perässä ei kuitenkaan näytetä niiden summia, vaan summat esitetään kummankin erän lopussa nimikkeiden vastaavaa ja vastattavaa toimiessa vain otsikoina. (Kirjanpitolautakunta 2006, 4-14.) Tässä ohjelmassa välisummien lisäksi vastaavan ja vastattavan arvot esitetään kuitenkin välittömästi nimikkeiden perässä eikä kyseisten erien lopussa, koska se on helpompi toteuttaa. Kirjanpito-ohjelmassa pitäisi pyrkiä kuitenkin siihen, että esitys noudattaisi voimassa olevia tase- ja tuloslaskelmakaavoja sekä yleistä tapaa niiden esittämiseksi. Käyttäjällä pitäisi olla mahdollisuus muokata esitystavan ja kaavat itse haluamukseen. Tällainen käytäntö tekee kirjanpito-ohjelmasta mahdollisimman pitkäikäisen siinäkin tapauksessa, että sen ylläpito lakkaisi.

Monet kirjanpito-ohjelmat automatisoivat arvonlisäveron laskennan. Vaikka arvonlisäverotieto on myynti- ja ostolaskuissa, toimii kirjanpito-ohjelman laskenta vielä varmistena sen lisäksi, että tietojen syöttäminen nopeutuu. Ohjelmat voivat myös muodostaa alv-laskelman automaattisesti. Kirjanpitolaissa tätä ei kuitenkaan vaadita, eikä sitä toteuteta tässä ohjelmassa.

3 Kirjanpidon järjestelmä

Tässä luvussa esitellään hyvin yleisellä tasolla kirjanpidon perusterminologia, juoksevan kirjanpidon kirjausten tekeminen ja tilinpäätöksen johtaminen kirjauksista. Luca Pacioli esitti kahdenkertaisen kirjanpidon menetelmän jo 1494 matematiikan oppikirjassaan (Tomperi 2016, 17). Alun perin kirjanpitoa on pidetty käsin kynän ja paperin avulla. Tietokoneiden keksiminen mahdollisti kirjanpidon automatisoinnin. Molemmissa tavoissa käytetään kuitenkin samoja käsitteitä kuten tilit, debet ja kredit.

3.1 Kahdenkertainen kirjanpito

Kirjanpidon merkinnät tehdään tileille. Tilillä on kaksi puolta: debet ja kredit. Debet kuvaa rahan käyttöä ja kredit rahan lähdettä. Debetistä käytetään kirjauksissa synonyymiä per ja kreditistä synonyymiä an. Juoksevassa kirjanpidossa on kolmenlaisia tilejä: meno-, tulo- ja rahoitustilejä. Lisäksi tilinpäätöksessä käytetään tilinpäätöstilejä. Menot (menon lisäykset) kirjataan menotilien debet-puolille ja vastaava määrä joidenkin rahoitustilien kredit-puolille. Tulot (tulon lisäykset) kirjataan tulotilien kredit-puolille ja vastaava määrä joidenkin rahoitustilien debet-puolille. Menon ja tulon vähennysten kanssa toimitaan edellä kuvatun kannalta käänteisesti. (Tomperi 2016, 15–27.) Edellä kuvattujen kirjausten lisäksi voi olla myös rahoitustilien välisiä kirjauksia. Kirjauksesta voidaan tässä työssä käyttää myös nimitystä vienti ja useammasta toisiinsa liittyvästä kirjauksesta termiä vientilauseke.

Yhdessä liiketapahtumassa voi olla pelkästään yksi kirjaus yhden tilin debet-puolelle ja toisen tilin kredit-puolelle. Esimerkki tällaisesta vientilausekkeesta on per Pankki an Myyntisaamiset $1,24 \times S$, jossa myyjä saa ostajalta suorituksena $1,24 \times S$ euron velkansa laskun mukaan. Tätä liiketapahtumaa ennen on myyty jotain per Myyntisaamiset $1,24 \times S$ an Myynti S an Myynnin ALV-velka $0,24 \times S$, jossa on kaksi rahoitustiliä ja yksi tulotili. Vaikka edellä tulotilille ja rahoitustilille kirjattavat määrät eivät ole yhtä suuret, an ja per ovat rahamääräisesti yhtä suuret. Kahdenkertaisen kirjanpidon ideana onkin, että tilien debet-puolien yhteenlaskettu summa vastaa kredit-puolien summaa eli ne pysyvät jatkuvasti yhtä suu-

rina. Jos näin ei ole eivätkä tilit mene tasan, on tehty virhe kirjanpidossa ja sanotaan, että kirjanpito erottaa. Edellä kuvatut viennit voidaan esittää myös tiliristikona (Kuva 1), jossa ristikon vasemmalla puolella on debet ja oikealla kredit.

	Myynti- saamiset	Myyntit	Myyntin ALV-velka	Pankkitili
1.1. Myyntilasku	1240	1000	240	
15.1. Saadaan rahat				1240

Kuva 1 Vientilausekkeita tiliristikoina esitettynä

3.2 Tase ja tuloslaskelma

Tilinpäätös on edellä kuvatun juoksevan kirjanpidon määrämuotoinen yhteenveto. Siinä kultakin tililtä lasketaan debet-puolen ja kredit-puolen summat erikseen. Näiden erotus eli tilin loppusaldo merkitään sille puolelle tiliristikkoa, kumman summa on pienempi ja vastaava summa merkitään samalla jonkin tilinpäätöstilin (tase- tai tulostilin) eri puolelle. Näin tili menee aina tasan ja loppusaldo ohjautuu taseeseen tai tuloslaskelmaan. Jos erotus lisättiin vaikkapa myyntitilin debet-puolelle, merkitään sama summa liikevaihdon kredit puolelle. Lopuksi tulostililtä lasketaan loppusaldo ja se siirretään samalla tavoin tasetilille. Jos tasetilit menevät silloin tasan, on kirjanpito oikein tehty. Tilinpäätöksessä rahoitustilit ohjataan tasetileille ja meno- ja tulotilit tulostileille. (Tomperi 2016, 29-30.)

Tuloslaskelma edustaa koko tilikautta. Tase taas edustaa tilinpäätöspäivän tilannetta. Uuden tilikauden alussa tehdään tilinavaus, jolloin edellisen tilikauden tasetileille ohjattujen rahoitustilien saldot siirretään tiliristikon vastaaville puolille rahoitustilien alkusaldoiksi. Sama koskee tasetileille ohjautuneiden menotilien menojäännöksiä (Tomperi 2016, 16). Tilikauden voitto ohjataan kuitenkin avausviennissä jollekin muulle oman pääoman tilille. Tasetilit toimivat ikään kuin kirjanpidon muistina, jonka avulla siirrytään tilikaudesta toiseen.

3.3 Yhdenkertainen kirjanpito

Yhdenkertaisessa kirjanpidossa kukin rahasumma kirjataan vain meno- tai tulotilille, joten tasetta ei synny lainkaan. Verotuksen tekemisen kannalta yhdenkertaisessakin kirjanpidossa kannattaa käyttää useampia meno- ja tulotilejä.

Jos kirjanpito-ohjelma vaatii, että debet ja kredit eivät saa erottaa, on kahdenkertaisen kirjanpidon ohjelmalla käytettävä vähintään yhtä tasetiliä vastatilinä, mikäli halutaan pitää yhdenkertaista kirjanpitoa.

4 Tietojärjestelmätieteen ja ohjelmistotuotannon teoriaa

Tässä luvussa käydään lävitse ensin ohjelmistoprojektin eri vaiheita teoriassa. Sen jälkeen tarkastellaan, millaisia malleja ohjelmistotuotannon historiassa on käytetty eri vaiheiden yhdistelemiseen. Tämän luvun teoriaa sovelletaan myöhemmin käytännössä kirjanpito-ohjelman suunnittelussa ja toteutuksessa luvussa 5. Tässä luvussa esiintyvät esimerkit liittyvät havainnollisuuden vuoksi kirjanpitoon ja suunniteltavaan kirjanpito-ohjelmaan.

Tietojenkäsittelytieteen (computer science) osa-alueet tietojärjestelmätiede (information system science) ja ohjelmistotuotanto eli ohjelmistotekniikka (software engineering) käyttävät samoja menetelmiä, mutta tietojärjestelmätiede kattaa ohjelmistotuotannon teknologian lisäksi tietojärjestelmien sosiaaliset, organisaationaaliset ja inhimilliset ulottuvuudet. Tietojärjestelmät on alun perin hoidettu täysin manuaalisina ennen kuin toiminnot on osittain automatisoitu. Tietojärjestelmät sisältävät rajapinnan tämän manuaalisen ja automaattisen osan välillä. (Pohjonen 2002, 6-8.). Tässä työssä termejä ohjelmistotuotanto ja tietojärjestelmätiede käytetään vaihtoehtoisesti.

4.1 Ohjelmistoprojektin vaiheet

Ohjelmistoprojektin vaiheita ovat järjestyksessä yleensä ainakin määrittely, suunnittelu, ohjelmointi eli implementointi ja testaus sekä käyttöönotto ja ylläpito (Häkälä & Mikkonen 2011, 30). Määrittelyn ajatellaan vastaavan kysymykseen mitä ja suunnittelun kysymykseen miten. Ohjelmistotuotanto voidaan nähdä myös peräkkäisten dokumenttien tuottamisena, jossa edellisen vaiheen dokumentti toimii

seuraavan vaiheen spesifikaationa. Silloin mitä-kysymys voidaan esittää aina kunkin vaiheen spesifikaationa ja vaiheen lopputuloksen pitäisi olla vastaus miten-kysymykseen. (Haikala & Märijärvi 2006, 61-63.)

4.1.1 Vaatimusmäärittely

Vaatimusmäärittely on vaatimustenhallinnan ohella vaatimusten käsittelyn osa. Sen vaiheet ovat asiakkaiden (laajemmin sidosryhmien) ohjelmalle asettamien vaatimusten kartoitus, analysointi, dokumentointi ja validointi. Lopputuloksena saadaan toiminnallinen määrittely. Vaatimukset voivat olla toiminnallisia tai ei-toiminnallisia sekä reunaehtoja. Toinen tapa jakaa vaatimukset on esittää ne eri abstraktiotasoilla asiakas- ohjelmisto- ja teknisinä vaatimuksina. Asiakasvaatimukset ovat lähtöisin asiakkaalta ja esitetty asiakkaan kielellä. Ne toteutetaan ohjelmistovaatimusten avulla. Kun ohjelmistovaatimukset muutetaan teknisiksi vaatimuksiksi, laskeudutaan hierarkiatasolla suunnittelutasolta toteutustasolle. Jäljitettävyys edellyttäisi sellaista dokumentointia, että asiakasvaatimuksista päästäisiin toteutustasolle ja takaisin toiseen suuntaan. (Haikala & Mikkonen 2011, 61-68.) Asiakasvaatimukset esittää tässä opinnäytetyön tekijä. Suurin osa niistä tulee lainsäädännöstä ja hyvästä kirjanpitolavasta ja ne on esitetty luvussa 2. Jäljitettävyyttä ei toteuteta täydellisenä.

Esitetään seuraavaksi konkreettinen esimerkki asiakasvaatimuksesta. Kirjanpitolain 2 luvun 2 § mukaan *liiketapahtumat merkitään asian mukaan kirjanpitoileille (kirjaus)* (Kirjanpitolaki 30.12.1997/1336). Lain vaatimus on siis samalla asiakasvaatimus. Se voidaan esittää esimerkiksi seuraavan kaltaisena ohjelmistovaatimuksena: Käyttäjä syöttää tosittien tiedot lomakkeelle, jolta ne viedään tietokannan kahteen eri tauluun. Tämä taas voidaan muuntaa esimerkiksi seuraavan kaltaiseksi tekniseksi vaatimukseksi: HTML-lomakkeen tiedot viedään JavaScriptillä ja PHP:llä tehdyn tarkastuksen jälkeen PHP:n PDO-tietokantarajapintaa hyödyntäen MySQL-tietokantaan. Prosessi kulkee siis abstraktista yksityiskohtaisemmaksi.

Vaatimusmäärittelyjen kuvaamiseen on käytetty 1980-luvun puolivälistä asti käytötapauksia (use cases). Ne kuvaavat asiakasvaatimukset ohjelmistovaatimuksina. Käyttötapauskuvaus sisältää käyttötapauksen nimen, version, alkutilanteen

tuloehdot, lopputilanteen ja kuvauksen, miten alkutilanteesta päästään lopputilanteeseen (Haikala & Mikkonen 2011, 80-82). Ohjelman toiminnalliset vaatimukset esitetään tässä työssä käyttötapauksien avulla. Versioita tehdään vain yksi, joten versiohistoriaa ei merkitä. Poikkeuksia ei tässä työssä dokumentoida.

Kirjauksen tekemisen käyttötapaus voitaisiin esittää esimerkiksi seuraavasti:

- Nimi: Tehdään kirjaus.
- (Versiohistoria: 3.5)
- Tuloehdot: Käytössä on tilikausi, jota ei ole lukittu ja käyttäjä on painanut linkkiä, jolta pääsee kirjaussivulle.
- Kuvaus: Käyttäjä syöttää kenttiin vähintään tositenumeron, tositelajin, päiväyksen, yhden tilinumeron ja sille kirjattavan rahamäärän ja valinnaisesti tositeselitteen, linkin tositteeseen. Käyttäjä lisää tilinumeroita ja niille rahamäärät ja riviselitteet sekä painaa lopuksi tallenna tietokantaan -painiketta.
- Lopputulos = Käyttäjän syöttämät tiedot tallennetaan tietokantaan.
- Poikkeukset = Käyttäjä ei täyttänyt pakollisia kenttiä tai tiedoissa oli virheitä, joiden takia niitä ei voitu syöttää tietokantaa, jolloin näytetään virheilmoitus tai palataan sivulle, mutta automaattisesti syötetty tositenumero ei kasva.

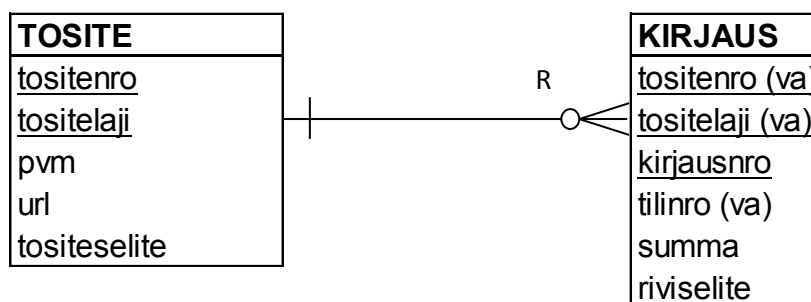
4.1.2 Suunnittelu

Suunnittelu jaetaan usein arkkitehtuurisuunnitteluksi ja moduulisuunnitteluksi. Arkkitehtuurisuunnittelu jakaa ohjelman moduuleihin ja keskittyy sitten niiden välisten ohjelmistorajapintojen (API) määrittelyyn. Moduulisuunnittelu taas keskittyy yksittäisiin moduuleihin ja niiden sisäiseen rakenteeseen. Suunnittelun tavoitteita ovat selkeys, ymmärrettävyys, luotettavuus, ylläpidettävyys, siirrettävyys ja tehokkuus. Keinoja näiden tavoitteiden saavuttamiseksi ovat esimerkiksi paikallisuus, modulaarisuus ja hierarkkisuus. Suunnittelussa otetaan kantaa ratkaisuperiaatteeseen, tietokanta-arkkitehtuuriin ja ohjelmistoarkkitehtuuriin. Suunnittelun lopputuloksena saadaan tekninen määrittely. (Haikala & Märijärvi 2006, 81-84.)

Tietokannan suunnittelu alkaa käsiteanalyysistä, jonka pohjana toimii vaatimusmäärittely. Sen tuloksena saadaan tietokantariippumaton käsitelmä, joka esite-

tään yleensä graafisena käsitekaaviona eli ER-kaaviona. Käsiteanalyysissä etsitään kohdealueen käsitteitä, joista halutaan säilyttää tietoa myöhempää käyttöä varten. Käsitteiden välillä voi olla yhteyksiä (verbejä). Käsite kuvataan laatikon sisällä sen otsikkona. Käsitteeseen liittyvät tiedot (ominaisuudet eli attribuutit) lisätään laatikkoon otsikon alle. Käsitteen perusavain eli pääavain on tieto, joka yksilöi käsitteen. Se alleviivataan ER-kaaviossa. Käsitteet, joiden välillä on yhteys, yhdistetään viivalla. Isä–lapsi-yhteyksissä lapsikäsitteen päähän piirretään harakanvarvas. Pakollisuus merkitään poikkiviivalla ja ehdollisuus nollalla. Viite-eheysäännöt voidaan myös merkitä kuvioon kirjaimilla. (Hovi, Huotari & Lahdenmäki 2005, 29-55.)

Esimerkiksi tässä työssä käsite-ehdokkaista ovat kirjanpitotili, tilinpäätöstili, kirjaus, tosite pääkirja ja tilinpäätös. Käsitteeseen kirjaus liittyvät tiedot tositenumero, tositelaji, tositekohtainen kirjausnumero, tilinumero ja kirjattava summa. Tositekäsitteen ominaisuuksia tässä ovat tositenumero, tositelaji, päivämäärä, tositteiden sijainti ja tositeselite (Kuva 2). Tässä työssä tositenumero ja tositelaji riittävät yksilöimään tositteiden. Kirjauksessa yksilöintiin tarvitaan niiden lisäksi tositekohtainen kirjausnumero. Havainnot, tositteesta tehdään ainakin kaksi kirjausta (kahdenkertaisessa kirjanpidossa) ja tilinpäätöstiliin kuuluu yksi tai useampi kirjanpitotili, ovat yksi–moneen-yhteyksiä eli isä–lapsi-yhteyksiä. Kirjauksen ne vierasavaimina toimivat tiedot, jotka toimivat tositteiden perusavaimina, yhdistävät käsitteet kirjaus ja tosite toisiinsa.



Kuva 2 ER-kaavio tositteiden ja kirjauksen välillä. Nimen lopussa oleva (va) tarkoittaa vierasavainta.

Ennen normalisointia tarveanalyysissä tarkennetaan käsittemallia lisäämällä tietoja käsitteisiin. Vaiheeseen kuuluu muun muassa lajittelukriteerien määrittäminen ja

hakuehtoien määrittely. Tietokannan normalisoinnissa minimoidaan tietojen toistaminen. Tavoitteena on tietojen eheyden ylläpito. Tarkastelu etenee normaalimuodosta yksi eteenpäin. Yleensä kolmas normalisointimuoto riittää. Normalisointi tarkastelee sarakkeiden eli attribuuttien välisiä riippuvuuksia kunkin käsitteen sisällä. Ensimmäiseen normaalimuotoon (1NF) päästään siirtämällä toistuvat ryhmät ja moniarvoiset sarakkeet (monta arvoa rivillä yhdessä sarakkeessa) omiin tauluihinsa eli muodostamalla niiden tiedoista uusi käsite. Taulu on toisessa normaalimuodossa (2NF), jos sen kaikki sarakkeet, jotka eivät ole perusavaimia ovat funktionaalisesti riippuvia (koko moniosasesta) perusavaimesta. Käsite, jolla on vain yksi perusavain, on siis aina toisessa normaalimuodossa. *Sanotaan, että sarake (tai kenttä) A on funktionaalisesti riippuva sarakkeesta (tai kentästä) B, jos jokaista B:tä kohti on aina korkeintaan yksi A:n arvo kunakin ajanhetkenä.* Normalisoinnissa sarakkeet, jotka eivät ole funktionaalisesti riippuvia koko perusavaimesta, muodostavat uudet käsitteet eli ne siirretään pois taulusta. Kolmannessa normaalimuodossa (3NF) kaikki sarakkeet ovat funktionaalisesti riippuvia vain perusavaimesta. Riippuvuus puretaan siirtämällä funktionaalisen riippuvuuden aiheuttava sarake, joka ei kuitenkaan toimi itse avaimena, omaan tauluun eli tekemällä siitä uusi käsite. (Hovi ym. 2005, 80-94.)

Tietokannan suunnittelun seuraavassa vaiheessa siirrytään tuoteriippumattomasta normalisoidusta käsitteellistä relaatiotietokannan tuotekohtaiseen taulurakenteeseen. Siirtymässä tietokannan taulu vastaa käsitettä, taulun sarake vastaa käsitteen tietoa ja viiteavaimet vastaavat käsitteiden välisiä yksi–moneen-yhteyksiä siten, että lapsitauluun tulee viiteavain. Taulut ja sarakkeet nimetään. Seuraavaksi määritellään sarakkeiden tietotyypit ja eheyssäännöt ja perustetaan perus- ja viiteavaimet. Tämä on jo tuotekohtaista, joten viimeistään tässä vaiheessa pitää ottaa kantaa tietokannan hallintajärjestelmään (TKHJ). Lopuksi toteutetaan poissulkevat yhteydet ja määritetään viite-eheydet. Vaihtoehtoja viiteeheyssäännöiksi on estä (restrict), vyörytä (cascade) ja nollaa (set null). Ne ottavat kantaa siihen, miten toimitaan, jos yritetään tehdä isä–lapsi-yhteyden eheyttä rikkova muutos tietokantaan. Viite-eheyssäännöt vaikuttavat myös taulujen perustamis- ja poistojärjestykseen. Lopuksi tietokannan suorituskykyä voidaan vielä virittää tietokantakyselyitä varten suunnittelemalla riittävät indeksit. (Hovi ym. 2005, 103-111.)

ER-kaavio voidaan esittää UML:n (Unified Modelling Language) luokkakaaviona. Siinä käsite on luokka ja yhteyksiä kuvataan assosiaationuolin. Nuolen suunta osoittaa isäkäsitteen. Nuolen päiden viereiset numerot kertovat lukumääräsuhteista. Tähti vastaa harakanvarvasta ja numero ennen pisteitä ehdollisuutta (0) tai pakollisuutta (1). (Hovi ym. 2005, 120-124.) Kuvan 2 ER-kaavio esitetään kuvassa 3 havainnollisuuden vuoksi taulumuodossa, jossa taulujen harmaat sarakkeetsikot osoittavat avainsarakkeet ja lyhenne va suluissa tarkoittaa vierasavainta. Taulukot on yhdistetty UML-notaation mukaisin nuolin, koska ne on helpompi piirtää kuin harakanvarpaat. Tässä työssä käytetään havainnollisuuden vuoksi totutun tavan vastaisesti kuvan 3 mukaista tapaa esittää ER-kaaviot. Liitteessä 2 käytetään kuitenkin luokkakaavioita tilan puutteen vuoksi.

Tositetaulu

tosite_no	tosite_laji (va)	päivä	tosite_url	tositeselite
1	1	1.6.2015		Myyntilasku 1
2	1	7.6.2015		Vuokranmaksu
3	1	13.6.2015		
1	2	4.7.2015	file://1.pdf	
...

↑ 1..1

Kirjaustaulu 0..*

tosite_no (va)	tosite_laji (va)	kirjaus_no	summa	tili_numero (va)	rivisel.
1	1	1	1240,00	1700	myyntis.
1	1	2	-1000,00	3000xx	itselle
1	1	3	-240,00	292040	ALV
2	1	1	500,00	701001	vuokra
2	1	2	-500,00	19100	
...

Kuva 3 Kuvan 2 mukaisen ER-kaavion esitys tauluina UML:n mukaisin nuolin

4.1.3 Toteutus ja testaus

Suunnitteluvaiheen jälkeen itse toteutus tehdään yleensä ohjelmoimalla ongelman ratkaisu käännettävällä tai tulkittavalla ohjelmointikielillä. Ohjelmointikieliä on eri abstraktiotasoilla suorittimien ymmärtämistä konekielistä hyvin korkean tason luonnollisen kielen kaltaisiin kieliin. Samoin ohjelmointiparadigmoja on monia

erilaisia. Tiedetyt kielet ja paradigmat sopivat paremmin tietyn tyyppisiin ongelmiin ja toiset toisen tyyppisiin.

Testaus nähdään suppeimmillaan suunnitelmallisena virheiden etsintänä ohjelmasta ja laajempaan kaikkina niinä toimenpiteinä, joiden tarkoituksena on mitata ja parantaa ohjelman laatua. Testauksella, jossa käytetään ohjelmaa, voidaan yleensä osoittaa, että ohjelmassa on virheitä, mutta sillä ei koskaan voida osoittaa, että ohjelma olisi virheetön. Tähän tarvittaisiin formaalia verifiointimenetelmää. Testauksessa ohjelman toimintaa peilataan toiminnalliseen ja tekniseen määrittelyyn. V-mallin mukaisessa testauksessa moduulitestaus testaa moduulisuunnittelua V:n kärjessä, integrointitestaus arkkitehtuurisuunnittelua V:n keskellä ja järjestelmättestaus määrittelyä V:n yläosassa. Mitä korkeammalla tasolla virhe huomataan, sitä enemmän aikaa sen korjaus vie. (Haikala & Märijärvi 2006, 284-291.)

4.2 Ohjelmistotuotannon mallit

Ohjelmistoprojekteista on esitetty erilaisia malleja, jotka sisältävät edellä kuvatut ohjelmistoprojektin vaiheet. Perinteisin lienee vuonna 1970 käyttöön otettu vesiputousmalli, alkaa vaatimusmäärittelyistä ja etenee suoraviivaisesti muiden vaiheiden kautta ohjelman käyttöönottoon. Väärinymmärrettynä siitä puuttuu iterointi takaisin päin. Vesiputousmallin hitauden vuoksi kehitettiin iteratiivisia menetelmiä, joissa vesiputouksia käydään projektin aikana lävitse useita. Tämä asetti vaatimuksia testauksen automatisoinnista ja dokumentaation tehostamisesta, kun sama työ jouduttiin tekemään moneen kertaan. Protoilussa taas ohjelmasta tehdään vaillinainen prototyyppi sen ominaisuuksien tutkimiseksi. Tultaessa 2000-luvulle alettiin ottaa käyttöön jo 1990-luvun vaihteessa kuvattuja ketteriä (agile) menetelmiä. Niissä asiakkaan tyytyväisyys ja toimiva ohjelmisto asetettiin byrokraattisten prosessien, dokumentaation ja työkalujen edelle. Näistä tunnetuin lienee erityisesti pieniin projekteihin sopiva Scrum, joka etenee noin kuukauden pyrähdyksinä, joiden aikana tiimi on rauhoitettu vaatimusmuutoksilta. Lean ja Kanban ovat ajattelutapoja, jotka on omaksuttu ohjelmistotuotantoon 2000-luvulla. Lean pyrkii muun muassa olemaan ihmiskeskeinen ja eliminoimaan kaiken työn, joka ei tuota asiakkaalle lisäarvoa. Kanbaniin kuuluu kolme periaatetta: työn

kulun visualisointi, yhtäaikaisten töiden määrän rajoittaminen ja läpimenoaikojen mittaaminen. (Haikala & Mikkonen 2011, 37-55.)

Menetelmästä riippumatta keskimäärin yli puolet elinkaarikustannuksista syntyy ylläpitovaiheessa. Yleinen havainto, että mitä pidemmälle projektia edetään, sitä suuremmaksi nousee uudelleen tehdyn työn osuus. Siksi on erityisen tärkeää pyrkiä välttämään virheitä varsinkin alkuvaiheessa eli vaatimusmäärittelyssä ja suunnittelussa. Jos vaatimusmäärittelyssä nimittäin uudestaan tehdyn työn osuus jää noin neljännekseen, tehdään koodausvaiheessa jo puolet ajasta korjauksia ja testausvaiheessa ennen käyttöönottoa jo lähes kaksikolmasosa työstä on uudestaan tekemistä. (Haikala & Märijärvi 2006, 56-58.)

5 Case kirjanpito-ohjelman suunnittelu ja toteutus

Kirjanpidon järjestelmä, kirjanpitolainsäädäntö, hyvä kirjanpitolain ja ohjelmistotuotanto muodostavat työn viitekehyksen. Ensimmäiseen tutkimuskysymykseen lainsäädännön ja hyvän kirjanpitolain asettamista vaatimuksista kirjanpito-ohjelmalle vastattiin luvussa 2. Vastuksen keskeisimmät kohdat on vielä esitetty taulukossa 1. Tämä luku vastaa pitkälti toiseen tutkimuskysymykseen: Miten ensimmäisen tutkimuskysymyksen vastausten mukainen kirjanpito-ohjelma suunnitellaan ja toteutetaan käytännössä? Tämä luku kuvaa vain yhden mahdollisen tavan toteuttaa kirjanpito-ohjelma luvussa 4 kuvatuin menetelmin, toisin sanoen juuri sen, jolla ohjelma toteutettiin tässä tapauksessa. Tehtyjä valintoja peilataan muihin mahdollisiin suunnitteluvaihtoehtoihin. Aina ei voi sanoa, että jokin valinta olisi yksiselitteisesti paras. Käyttäjälle samannäköiseen lopputulokseen voitaisiin päätyä monella eri suunnittelu- ja toteutusratkaisulla.

Termejä käyttäjä ja kirjanpito-ohjelman käyttäjä käytetään jatkossa rinnan. Tässä työssä toteutetussa kirjanpito-ohjelmassa ne ovat kuitenkin tarkkaan ottaen eri asioita. Kirjanpitäjällä (ohjelman käyttäjä) voi olla asiakkaana monta kirjanpito-ohjelmasta. Toisaalta tässä ohjelmassa kukin kirjanpito-ohjelman on oma käyttäjänsä (erillinen käyttäjätunnus). Kirjanpitäjä näkee siis yhdellä sisäänkirjautumisella vain yhden kirjanpito-ohjelman tiedot.

Kirjanpitolain vaatimukset kirjanpito-ohjelmalle	
Juokseva kirjanpito	Tilinpäätös
<ul style="list-style-type: none"> - Kirjaus perustuu liiketapahtumaan liittyvän tositteeseen sisältöön. - Kirjaus vaatii suoraan kirjanpilotilin numeron tai selityksen (nimen). - Jäljitettävyyden luo tositteeseen ja kirjauksen välille vaatimuksen päiväyksestä tai tositenumerosta tai muusta yksilöivästä tiedosta. - Kirjaukseen pitää liittää rahamäärä, vaikka laki ei sitä eksplisiittisesti edellytä. - Kirjauksia on voitava tarkastella aika- ja asiajärjestyksessä. - Pääkirjanpidosta on oltava kaksisuuntainen katkeamaton ketju (audit trail) sekä tosittesiin, että tilinpäätökseen. 	<ul style="list-style-type: none"> - Tilinpäätös on tehtävä suomen tai ruotsin kielellä euro-määräisenä. - Kirjanpidon tileistä on muodostettava tililuettelo. - (Tuloslaskelman ja taseen vertailutiedot on oikaistava tarvittaessa.) - (Tilinpäätöstä varmentamaan on laadittava tase-erittelyt ja liitetietojen erittelyt) - (Kirjanpitoaineistojen lajeista pitää olla luettelo.)
PMA-asetuksen vaatimukset ohjelmalle ja KILA:n ohjeiden vaatimukset	
<ul style="list-style-type: none"> - PMA-asetuksessa esitetään noudatettavat tuloslaskelmakaavat, tasekaava ja liitetietovaatimukset. - Kaavat voidaan esittää annettua tarkemmin ja se voi olla vaatimuskin joissain tilanteissa. - Nimikkeitä ei merkitä tulolaskelmaan tai taseeseen, jos nimikkeen kohdalle ei tule lukua tilikaudelle tai vertailuvuodelle. - Tuloslaskelmasta ja taseesta voidaan jättää etusarakkeet pois sekä näyttää tilalla tietyt välisummat. (KILA) 	

Taulukko 1 Tiivistelmä tutkimuskysymyksen 1 vastauksista

5.1 Ohjelman vaatimusmäärittely

Tässä luvussa käydään aluksi läpi niitä kirjanpito-ohjelman vaatimuksia, joita ei ole jo käsitelty luvussa 2. Vaatimukset koskevat siis muita asioita kuin mahdollisuutta tuottaa kirjanpito-ohjelmalla lainmukainen kirjanpito ja tilinpäätös. Nämä vaatimukset ovat pitkälti työn tekijän asettamia toiminnallisia ja ei-toiminnallisia vaatimuksia sekä reunaehtoja.

Tässä työssä toteutettavan ohjelman pitää ensinnäkin olla käytettävissä mahdollisimman monella erilaisella laitteella älypuhelimista tietokoneisiin ja sen käyttöönoton on oltava mahdollisimman helppoa. Jos vaadittaisiin, että ohjelma tulee toteuttaa selainkäyttöisenä, asetettaisiin suoraan ohjelmistovaatimus. Edellä esitetty vaatimus laiteriippumattomuudesta on kuitenkin asiakasvaatimus, joka vahvasti puoltaa selainkäyttöistä käyttöliittymää. Sopivilla asiakasvaatimuksilla päästään siis haluttuihin ohjelmistovaatimuksiin. Jatkossa vaatimusmäärittelyssä ajatellaan, että on suoraan asetettu reunaehto siitä, että tämä ohjelma pitää toteuttaa selainkäyttöisenä ja sitä on voitava käyttää internetissä ilman, että ohjelmaa asennetaan käyttäjän laitteelle.

Työn tekijällä olemassa olevan webhotellipalvelun vuoksi ohjelman on oltava asennettavissa käytettäväksi webhotelliin jaetulle palvelimelle. Jaetun palvelimen hallinta on melko rajoitettua verrattuna kalliimpiin virtuaalipalvelimiin tai omiin palvelimiin. Webhotellissa on yleensä rajoitettu määrä dataliikennettä ja säilytystilaa. Toiminnallisena vaatimuksena esitetään ohjelman webhotelliinsa asentavalle aiheutuvien käyttökustannusten minimointia. Ensisijaisesti tässä halutaan minimoida siirrettävän tiedon määrä ja sen jälkeen tallennettavan tiedon määrä.

Koska tämä ohjelma on käytettävissä internetissä, tulee sen voida olla useamman kirjanpitovelvollisen käytössä. Uuden kirjanpitovelvollisen lisääminen on automatisoitava. Muut kirjanpitovelvolliset eivät kuitenkaan saa nähdä toistensa tietoja eikä ulkopuolisilla tule olla pääsyä kirjanpitovelvollisen tietoihin. Käyttäjistä ei pidä kerätä tarpeettomasti henkilötietoja tai muita tietoja. Henkilötietolain noudattaminen jätetään kirjausten osalta käyttäjän vastuulle, samoin ohjelman ulkopuolisten dokumenttien (tositteiden) tallentamispaikan osalta.

Jo lainsäädännöstä tulee vaatimus tarvittavan yksityiskohtaiseen kirjanpitoilien käyttöön. Tässä toteutettavassa kirjanpito-ohjelmassa käyttäjän pitää voida lisätä omia kirjanpitoilejä myös siksi, että hän voi toteuttaa kustannuspaikkaseurantaa. Käyttäjälle tullaan näyttämään valmiina annetut ja hänen itse lisäämänsä kirjanpitoilit, joista valinnat tehdään kirjauksia tehtäessä. Valmiina annetun tilikartan tulee noudattaa jossain määrin raportointikoodiston mallia, jotta verotusta varten myöhemmin mahdollisesti toteutettava XBRL-raportointi on helpompi suunnitella ja toteuttaa.

Kirjanpitovelvollisella pitäisi lain mukaan olla mahdollisuus antaa tilinpäätös minimaatimuksia yksityiskohtaisempana, jos se on tarpeen oikean ja riittävän kuvan saamiseksi. Käyttäjän pitäisi voida muokata muutoinkin tilinpäätöstä sellaiseksi kuin haluaa. Tässä toteutuksessa ei kuitenkaan anneta käyttäjälle mahdollisuutta lisätä tilinpäätöstilejä eikä muokata tuloslaskelman ja taseen rakennetta.

Koska käyttäjä saattaa tehdä virheitä, tulee hänellä olla myös mahdollisuus muuttaa tai poistaa itse tekemiään kirjanpitotilejä. Samoin käyttäjän tulee voida korjata tositteeseen liittyvät virheelliset kirjaukset ja poistaa kaikki tositteeseen liittyvät kirjaukset. Kirjanpitovelvollisen pitää voida ottaa varmuuskopiot omista tiedoistaan ja tallentaa ottamansa varmuuskopiot takaisin ohjelmaan.

Käyttäjän pitää saada lisätä tositenumeron lisäksi tositteen tositelajin numero. Tositelaji voi olla vaikka palkkakirjanpito tai ostotosite. Tämä mahdollistaa esimerkiksi ostoreskontran tuonnin myöhemmin tai lajittelun tositelajin perusteella. Tositelaji on kaikissa tositteissa pakollinen tieto. Tositelajinumeroa yksi käytetään oletustositelajina, mikäli ei anneta muuta tositelajia. Tälle ohjelmalle ei aseteta vaatimusta debet- ja kredit-puolien yhtäsuuruudesta, jolloin yhdenkertaista kirjanpitoa voidaan pitää ilman keinotekoisia tasetiliä. Tämä ei estä toteuttamasta huomautusta tilien erottamisesta. Huomauttaminen toteutetaan.

Tässä mainitaan vielä erikseen selvyiden vuoksi joitakin asioita, joita kirjanpito-ohjelmassa ei tässä työssä toteuteta. Tällaisia asioita ovat alv-automatiikka, osakirjanpidot, tuki muille kielille kuin suomen kieli ja käyttäjän mahdollisuus muokata tilinpäätöstä suoraan ohjelman avulla. Muiden kuin liiketoimintaa harjoittavien mikroyritysten tuloslaskelma- ja tasekaavoja ei toteuteta. Tällaisia ovat esimerkiksi aatteellisen yhteisön- ja säätiön tuloslaskelmakaava sekä kiinteistökaava ja toimintokohtainen tuloslaskelma.

Käyttötapauskuvaukset esitetään erikseen liitteessä 1. Käyttötapauksista esitetään versio, joka vastaa lopputulosta. Liite käykin ohjelman ohjekirjasta. Seuraavaksi luetellaan merkittävimmät käyttötapauskuvat. Käyttäjä syöttää tositteen tiedot ja tallettaa ne tietokantaan. Käyttäjä korjaa tositteen tietoja tai poistaa tositteen tiedot. Käyttäjä kirjaa liitetiedon tai muuttaa sitä. Käyttäjä lisää oman kirjanpitotilin tai muokkaa sitä tai poistaa sen. Tietojen syöttämisen jälkeen käyttäjä tarkastelee

näytöllä tilinpäätöstä tai pää- tai päiväkirjaa. Käyttäjä lataa tase-erittelyt jatkojaloistamista varten. Käyttäjä avaa uuden tilikauden, sulkee tilikauden tai vaihtaa tilikautta.

5.2 Ratkaisuperiaatteen ja ohjelmistoarkkitehtuurin suunnittelu

Tässä luvussa kuvataan yleisiä seikkoja ja ratkaisuperiaatteita, jotka on huomioitu tämän ohjelman suunnittelussa ja myöhemmin toteutuksessa. MVC-malli pyrkii erottamaan mallin (model), käyttöliittymän eli näkymän (view) ja ohjauksen (controller) toisistaan. Ohjelmassa ei noudateta MVC-mallia, mutta käyttöliittymä pyritään pitämään erillisenä sovelluslogiikasta.

Ohjelmasta käyttäjälle näkyvä osuus (front end) päätettiin toteuttaa HTML-koodilla toteutetuista verkkosivuista, sillä verkkosivuja voi käyttää selaimella, eikä se yleensä vaadi käyttäjältä mitään asennuksia omalle laitteelle. Näin saadaan tuki mahdollisimman suurelle määrälle laitteita ja toteutetaan helppo käyttöönotto vaatimusmäärittelyn mukaisesti. Toisaalta selainten tavassa tulkita HTML-koodia on eroja, joten toiminnallisuudet saattavat erota selainten välillä verrattuna tilanteeseen, että käyttöliittymä olisi ohjelmoitu erilliseksi ohjelmaksi.

Jos ohjelmassa käytetään HTML-koodin lisäksi JavaScript-koodia, voidaan käyttäjän selaimessa suorittaa tarkastuksia ja lisätä verkkosivuille dynaamisia elementtejä. Yksi asiakasvaatimuksista oli siirrettävän tiedon määrän minimointi. Sijoittamalla tilikartta ulkoiseen JavaScript-tiedostoon voidaan palvelimelta haettavan tiedon määrää vähentää. Silloin käyttäjän on kuitenkin sallittava JavaScriptin suorittaminen selaimessaan, jos hän haluaa nähdä tilikartan tai hyödyntää muita dynaamisia ominaisuuksia. Ajax-sovelluskehitystekniikoilla voitaisiin vähentää kirjauksien yhteydessä tapahtuvaa tilikartan siirtelyä. Ajax-tekniikoita ei kuitenkaan tulla käyttämään. Selainten tavassa tulkita myös JavaScript-koodia on eroja, joten toiminnallisuudet saattavat erota selainten välillä. Ohjelma päätettiin toteuttaa niin, että se toimii ilman JavaScriptin sallimista selaimessa, mutta käytettävyys huononee, jos JavaScript ei ole käytössä.

Käyttäjälle näkymätöntä osa (back end) vastaa sovelluslogiikasta. Esimerkiksi käyttäjän antamat kirjanpitoon liittyvät tiedot täytyy tallentaa jonnekin, jotta ne

ovat käytössä sivunlatausten välillä, sillä HTML on tilaton protokolla. Tiedot voidaan tallettaa muun muassa käyttäjän koneelle selaimen välimuistiin, samalle palvelimelle, jossa ohjelma toimii tai ulkopuoliselle palvelimelle. Tallentaminen käyttäjän koneelle selaimen välimuistiin on osin selainriippuvaista ja siinä on vaarana, että käyttäjä menettää tiedot tyhjentäessään selaimen välimuistin. Koska tallennettavien tietojen hallinnointi on helpointa ja turvallisinta palvelimella, josta verkkosivut jaellaan, päädyttiin tallentamaan tiedot samalle palvelimelle kuin ohjelmakin. Tietokantaan tietojen säilytyspaikkana suoran tiedostonkäsittelyn sijasta päädyttiin samasta syystä. Luvussa 5.3 keskitytään erikseen tietokannan suunnitteluun.

Back end muodostuu tietokannan lisäksi jostain palvelinpuolen käännettävästä tai tulkittavasta ohjelmointikielestä. Tällaisia ovat esimerkiksi PHP, Python, Perl ja Java, joista päätettiin käyttää PHP:tä, koska se on yleisin ja parhaiten tuettu webhotelleissa. PHP-kielellä toteutetaan sivun toiminnallisuus, tarkastetaan käyttäjän syötteet ja tallennetaan ne tietokantaan. Back endin ja front endin erottaminen toisistaan mahdollistaa sen, että tietokannassa olevia tietoja voitaisiin käsitellä verkkosivujen lisäksi vaikkapa erillisellä käyttäjän laitteelle asennettavalle ohjelmalla. Kaikessa suunnittelussa pyritään noudattamaan tällaista modulaarisuutta.

Koska ohjelman on tarkoitus olla käytettävissä internetissä, pitää SQL-injektioiden ja muiden haavoittuvuuksien välttämiseen kiinnittää paljon enemmän huomiota kuin omalle laitteelle asennettavan ohjelman tapauksessa. Koska haavoittuvuuksia ohjelmaan jää varmasti, pitää tietoja kerätä käyttäjistä mahdollisimman vähän. Tunnistustiedot ja arkaluonteiset tiedot voidaan siirtää salattuna ostamalla SSL-sertifikaatti tai tekemällä oma sertifikaatti. Se kuuluu kuitenkin tämän työn ulkopuolelle.

5.3 Tietokannan suunnittelu

Tässä luvussa keskitytään erityisesti tietokannan suunnitteluun. Ensin käydään lävitse yleistä tietokanta- ja taulurakennetta ja sitten yksittäisiä tauluja. Kaikkiin tauluihin tallennetaan tietokannanhallintajärjestelmän avulla aikaleima automaattisesti ilman, että siitä mainitaan jatkossa. Taulujen suunnittelun yhteydessä ei

erikseen käsitellä niiden tietotyyppejä, mikäli sen voi olettaa olevan ilmeinen. Käytännössä suunnittelussa ei muodostettu luvussa 4 kuvattua käsittemallia lainkaan, vaan ensimmäinen graafinen esitys tietokannalle oli normalisoitu käsittemalli suoraan tauluina kuvattuna. Siksi luvun esimerkeissä kuvataan mallia oleellisilta osin tauluilla. Tauluista ja tekstistä saattaa puuttua sarakkeita, joita käytetään lopullisessa ohjelmassa mutta joiden merkitys on vähäinen. Lopullisesta valmiista tietokannasta on piirretty luokkakaavio liitteessä 2. Siitäkin puuttuu kaikissa tauluissa olevat aikaleimat.

Kaikkien ohjelmaa käyttävien kirjanpitovelvollisten tiedot voidaan tallentaa samaan tietokantaan tai jokaiselle kirjanpitovelvolliselle voidaan luoda oma tietokanta. Koska osa hosting-palvelujentarjoajista myy halvimmissa käyttäjien kesken jaetuissa webhotellituotteissa rajoitetun määrän tietokantoja tai vain yhden tietokannan, päätettiin kaikkien kirjanpitovelvollisten kirjanpidot käsitellä samassa tietokannassa, vaikkakin erilliset tietokannat parantaisivat tietoturvaa. Lisäksi tietokantojen luomisen ja käyttöoikeuksien lisäämisen saattaa joutua tekemään manuaalisesti, jolloin uuden kirjanpitovelvollisen lisäämistä ei voisi automatisoida, jos käyttäjät olisivat eri tietokannoissa. Tällaista ongelmaa ei esiintyisi kalliimmilla omilla virtuaalipalvelimilla.

Yhden kirjanpitovelvollisen tietokantatauluihin voidaan tallettaa monien tilikausien tiedot siten, että kirjanpitovelvollisen tauluihin pitäisi lisätä sarake, joka kertoo, minkä tilikauden tiedosta on kyse. Toinen vaihtoehto on käyttää kullekin tilikaudelle omia tauluja, joiden nimeämisessä on huomioitu tilikausi. Vaikka tilikausisarake viekin tilaa tauluista, vie suurempi määrä tauluja todennäköisesti enemmän tilaa, mikäli vuosittaiset kirjausten määrät jäävät pieniksi. Tilikausisarakkeen käyttö vaatii ylimääräisen WHERE-ehdon tietokantakyselyissä, mutta yhden taulun käsittelyn ohjelmointi on helpompaa kuin tilikausikohtaisten taulujen käsittely, koska tilikautta ei tarvitse huomioida taulujen nimissä. Silloin tauluja ei myöskään tarvitsisi nimetä uudelleen tai kopioida tilikauden vaihteessa. Koska haku lyhennästä tauluista on kuitenkin nopeampaa, ei tilikausisaraketta otettu käyttöön, vaan päätettiin toteuttaa tilikauden mukaisesti nimetyt taulut. Tähän vaikutti myös se, että ohjelmaa oli toteutettu jo tilinpäätökseen asti ennen kuin tilikauden vaih-

tamista alettiin suunnitella. Yhteiset taulut tilikausille olisi ollut myös käyttökelpoinen vaihtoehto. Opinnäytetyön tekijän havaintojen mukaan esimerkiksi Pupesoft-toiminnanohjausjärjestelmässä ei nimetä tauluja tilikauden mukaan, vaan käytetään tauluissa tilikausisarakea.

Kaikki kirjanpitovelvolliset eli tässä toteutuksessa eri käyttäjien tiedot voidaan tallentaa samoihin tauluihin lisäämällä käyttäjäsarake kaikkiin tauluihin, kuten edellä pohdittiin tilikausien kanssa meneteltävän. Toinen vaihtoehto on käyttää käyttäjäkohtaisia tauluja. Samoista syistä kuin edellä ja yksityisyys- sekä tietoturvasyistä eri kirjanpitovelvollisten tiedot päätettiin tallentaa eri tauluihin. Yhteisten taulujen käyttö kaikille kirjanpitovelvollisille olisi kuitenkin vähintään yhtä hyvä ratkaisu. Esimerkiksi Pupesoftissa on opinnäytetyön tekijän havaintojen mukaan päädytty ratkaisuun, jossa eri yritykset ovat samoissa tauluissa. Kuten edellä mainittiin, olisi käyttäjäkohtainen tietokanta kuitenkin turvallisempi ratkaisu. Silloin tietoturva voitaisiin hoitaa tietokannan käyttäjähallinnalla eikä SQL-injektio olisi mahdollinen ilman käyttäjän salasanan selville saamista.

Luontainen valinta on siis käyttää joko tilikausikohtaisia ja käyttäjäkohtaisia tauluja molempia erikseen tai hallita molempia – tilikautia ja käyttäjiä – sarakkeiden avulla samoissa tauluissa. Kolmas mahdollisuus olisi käyttäjäkohtaiset taulut, joissa tilikaudet olisivat sarakkeissa. Se toisi muunneltavuutta, jos eri käyttäjät haluttaisiin myöhemmin siirtää omiin tietokantoihinsa. Tilikausikohtaiset taulut käyttäjäsarakein eivät tule kyseeseen, sillä eri yritysten tilikaudet voivat vaihtua eri aikaan.

Tilikausikohtaisiin tauluihin päästään muun muassa tekemällä aina uudet taulut eri nimillä tilikauden vaihteessa tai muuttamalla käytössä olleiden taulujen nimiä tilikauden lopussa ja käyttämällä samannimisiä tauluja kullakin tilikaudella. Ohjelmassa päädyttiin ratkaisuun, jossa uudemman tilikauden taulut kopioidaan vanhemman tilikauden taulujen päälle vanhimmasta tilikaudesta alkaen. Aluksi vanhimman tilikauden taulut kuitenkin kopioidaan toisille nimille. Tämä mahdollistaa tilikausien käsittelyn vakionimisissä tauluissa ja toiminnan seurauksena suurimmalla numerolla nimetyt taulut edustavat vanhimpia tilikautia.

5.3.1 Pääkirjanpidon taulut

Vaikka paperisessa kirjanpidossa pääkirjan tilit ovat suurimmilta osin eri tiliristikoissa, voidaan tietokannassa kaikki tilit pitää samassa taulussa, sillä tilien numeroiden läpikäynti on nopeaa. Tämä vastaa paperisen kirjanpidon erottelutiliä. Jos jokaista tiliä varten tehdään oma taulunsa, kasvaa tietokannan taulujen määrä nopeasti hyvin suureksi. Tilikohtaisten taulujen läpikäynti olisi nopeaa pääkirjaa ja tuloslaskelmaa muodostettaessa mutta taulujen hallinnointi ja päiväkirjan muodostaminen hankalaa. Kaikilla tileillä ei välttämättä olisi montaa kirjausta, mutta ne kuluttaisivat silti turhaan levytilaa. Tässä päädyttiin pitämään kaikki tilit samassa taulussa kuten erottelutilillä eikä muodostettu tiliristikoille omia tauluja.

Kirjanpito-ohjelmassa pitää olla rahamäärien tallettamista varten debet-tilit ja kredit-tilit tai debet-sarake ja kredit-sarake tai yksi taulu tai sarake, jossa debet ja kredit erotetaan toisistaan jollakin muulla tavoin. Kredit-puolen arvot voidaan merkitä negatiivisina ja debet-puolen arvot positiivisina tietokantaan, vaikka käyttäjän ei tarvitsisikaan lomakkeella syöttää tietoja etumerkin kanssa. Näin tietokannassa tarvitaan yksi taulu tai sarake vähemmän. Tässä työssä päätettiin käyttää vain yhtä taulun saraketta rahamäärien summien tallentamiseen, siten että kredit merkitään tietokantaan negatiivisena ja debet positiivisena lukuna kahden desimaalin tarkkuudella.

On hyvä huomata, että käyttäjän näkemä käyttöliittymä ja takana oleva tietokantaratkaisu saattavat hyvinkin erota toisistaan. Päiväkirja ja pääkirja pitää saada näkyville, mutta niitä ei tarvitse säilyttää tietokannassa sellaisenaan ja vielä erikseen. Sisällöltään identtiset mutta järjestykseltään erilaiset päiväkirjanpitotaulu (Kuva 4) ja pääkirjanpitotaulu kannattaa normalisoida kolmanteen normaalimuotoon. Silloin tietoja ei toisteta sarakeissa turhaan. Jos samalta tosittelulta nimitäin tehtäisiin useampi kirjaus, mikä kahdenkertaisessa kirjanpidossa tehdään aina, jouduttaisiin toistamaan päivämäärä-, tositieselite-, tositenumero-, tosittelaji- ja linkkisarakkeiden arvot eri kirjauksille, kuten kuvasta 4 nähdään.

Päiväkirjanpitotaulu

t_no	t_laji	kirj_no	päivä	url	selite	summa	tili	riviselite
1	1	1	1.6.15	X	myynti	1240,00	1700	myyntisaaminen
1	1	2	1.6.15	X	myynti	-1000,00	3000xx	itselle
1	1	3	1.6.15	X	myynti	-240,00	292040	alv-velka
2	1	1	7.6.15	Y	vuokra	500,00	701001	
2	1	2	7.6.15	Y	vuokra	-500,00	19100	tililtä 1
...
1	2	1	5.6.15	Z	palkka	950,00	50010	palkka
1	2	2	5.6.15	Z	palkka	-950,00	19100	tililtä 1
...

Kuva 4 Päiväkirjanpitotaulu ennen normalisointia

Normalisoinnilla saadaan kaksi eri taulua: tositem- ja kirjaustaulu (Kuva 3 sivulla 20). Yhden kirjaustaulun käyttö toisi nopeutta hakuihin eli pääkirjan ja päiväkirjan tarkasteluun kahteen tauluun verrattuna. Toistettavan tiedon määrä ei vielä kasvaisi merkittävästi nyt suunnitellusta ratkaisusta. Jos koko tositem haluttaisiin myöhemmin kuitenkin tallentaa tietokantaan pelkän linkin sijasta, lisääntyisi tilan hukkaaminen merkittävästi. Tämä tukee kirjausten jakamista erikseen tositem- ja kirjaustauluihin.

Tositetaulun perusavaimena toimii tositemnumero ja tositemlaji yhdessä. Muita tauluun tallennettavia sarakkeita (kenttiä) ovat päiväys ja linkki tositemeseen (Kuva 3). Jos tositemlajille haluttaisiin antaa nimi, pitäisi tositetaulun tositemlajisarakkeen olla viiteavain tositemlajitaulun tositemlajin numeroon. Tositemlajitaulussa olisi toinen sarake, tositemlajin nimi. Tositemlajitaulua ei toteutettu, vaan käyttäjän velvollisuudeksi jää huolehtia, mitä kukin tositemlajin numero tarkoittaa.

Kirjaustaulun tositemnumero, tositemlaji ja kirjauksen numero muodostavat moniosaisen perusavaimen kahden ensimmäisen toimiessa viiteavaimena tositetauluun (Kuva 3). Lisäksi kirjaustauluun tallennetaan pakollisina tietoina summa eli rahamäärä (debet positiivisena ja kredit negatiivisena) ja tilinumero. Tilin nimeä ei tallenneta kirjaustauluun, sillä tilinumero on viiteavain tilin nimen sisältävään kirjanpitotilien tauluun. Riviselite ei ole pakollinen tieto. Annettaessa se tallennetaan omaan sarakkeeseensa. Eheyssäännöksi tilinumerolle käy rajoita (restrict),

sillä tilinumeroa ei saisi poistaa tai muuttaa, niin kauan kuin on olemassa kirjauksia, jossa tilinumeroa käytetään. Koska yhteiset tilinumerot voivat olla muidenkin käytössä, niin vierittämissäntöä (cascade) eli avainten arvojen muuttamista ei voida tässä käyttää. Tositenumeron ja -lajin eheysäännöksi käy kumpi vaan vierittäminen tai rajoittaminen tietojen päivityksen yhteydessä. Tässä valitaan vierittäminen. Toteutuksessa sillä ei ollut merkitystä, sillä kirjauksia muutettaessa vanha kirjaus poistettiin ja tilalle vietiin aina uusi muutettu.

Tosite- ja kirjaustaulujen (sekä kirjanpitotilien taulun) liitoskysely muodostaa lain vaatiman pääkirjanpidon. Liitoskysely on kysely, joka yhdistää isä- ja lapsitaulun liittämällä lapsitaulun viiteavaimen isätaulun vastaavaan avaimeen. Liitoksen lajittelu aikajärjestykseen tuottaa päiväkirjan ja lajittelu tilinumeroiden mukaan tuottaa asiajärjestyksen eli pääkirjan.

5.3.2 Kirjanpitotilien taulu (tilikartta)

Kirjanpitotilien taulun (Kuvassa 5 keskellä) perusavain on kirjanpitotilin numero. Se toteutetaan merkkijonotyyppisenä toisin kuin nimi antaisi ymmärtää. Numero-toteutuksessa voitaisiin tilinpäätöstä tehtäessä hakea kirjanpitotilien tilinumerot tarvittavalta väliltä. Esimerkiksi kaikkia numeroiden 10000 ja 29999 välillä olevat tilit ohjattaisiin taseeseen ja numeroiden 10000 ja 10999 välillä olevat tilit taseen aineettomiin hyödykkeisiin. Tämä onnistuu helposti numeroille, mutta ei merkkijonoille ilman muunnosta numeroksi. Tämä edellyttäisi kuitenkin, että kaikkien tilinumeroiden pituus olisi sama. Jos halutaan, että tilinumeroiden pituus saa vaihdella, kannattaa tilinumerosta tehdä merkkijonotyyppinen muuttuja. Silloin kirjanpitotilit voidaan ohjata tilinpäätöstileille tilinumeron alun mukaan. Esimerkiksi liikevaihto laskettaisiin etsimällä numerolla kolme alkavat kirjanpitotilit ja laskeamalla niiden kirjausten rahamäärien summa. Kirjanpitotilien taulussa on myös sarake tilin nimelle ja tilin tyyppi. Kirjanpitotilien tauluun lisätään sarake emotili, joka on viiteavain tilinpäätöksen ohjaustilien tilinumerokenttään. Tieto kertoo sen, minkä tilinpäätöstilin alle kyseinen kirjanpitotili kuuluu tilinpäätöksessä. Emotilin eheysäännöksi luontainen valinta on muutosten ja poistamisen rajoittaminen, sillä hierarkia, johon kirjanpitotilin emotili viittaa, on kaikille käyttäjille yhteinen.

Tilinpäätöshierarkiataulu

tp_tili_no	tili_nimi	emotili_no	alku	loppu
...
028	Vieras pääoma	02	28	29
0280	Pitkäaikainen vieras pääoma	028	28	28
0290	Lyhytaikainen vieras pääoma	028	29	29
030	Liikevaihto	03	30	30
...

Tilikarttataulu

kp_tili_no	tili_nimi	tp_tili_no (va)
...
292030	Jäsenmaksutilitysvelka	0290
292040	Arvonlisäverovelka	0290
...
3001	Tavaroiden kotimaan myynti	030
3002	Palveluiden kotimaan myynti	030
...

Kirjaustaulu

tosite_no (va)	tosite_laji (va)	kirjaus_no	summa	kp_tili_no (va)	rivisel.
1	1	1	1240,00	1700	myyntis.
1	1	2	-1000,00	3000xx	itselle
1	1	3	-240,00	292040	ALV
2	1	1	500,00	701001	vuokra
2	1	2	-500,00	19100	
...

Kuva 5 Tilinpäätöksen hierarkiataulun, kirjanpitolilien taulun ja kirjaustaulun väli-set suhteet

Tilikaudella käytettyjen kirjanpitolilien luettelo saadaan kirjaustaulun ja kirjanpito-tilien taulun liitoksena. Se päätettiin tallentaa omaan tauluunsa, joka on täysin identtinen kirjanpitolilien taulun kanssa, mutta sisältää vain ne tilit, joilla on vien-tejä. Tämä ei olisi välttämätöntä, jos kirjanpitolilien tauluun olisi lisätty sarake, joka kertoo onko tili ollut käytössä. Toinen vaihtoehto olisi muodostaa tarvittava liitos aina, kun haluttaisiin saada tieto käytetyistä tileistä.

5.3.3 Tilinpäätöksen taulut

Tilinpäätöstä varten kirjanpito-ohjelmassa on oltava tilien ohjaus. Esimerkiksi eri myyntitilit ohjautuvat liikevaihdon alle ja eri myyntitilien summa on liikevaihto. Tämä käsite vaatii oman hierarkiataulunsa (Kuvassa 5 ylinnä). Sen perusavaimena toimii tilinpäätöstilin tilinumero. Taseen ja tuloslaskelman yhteisen hierarkiataulun muita sarakkeita ovat tilinpäätöstilin nimi eli tilinpäätöksessä näytettävä nimike ja emotilin tilinumero, joka on arvoltaan NULL tai jokin hierarkiataulun perusavaimen arvoista. Emotili on ikään kuin taulun sisäinen viiteavain perusavaimeen, mutta ilman eheystarkastuksia. Sarake emotilin numero luo tauluun tilinpäätöksen puumaisena rakenteena kuvaavan hierarkian viittaamalla taulukon toiseen tilinpäätöstilin numeroon, perusavaimeen. Kirjassa Tietokantojen suunnittelu ja indeksointi on annettu yksityiskohtaisempi esimerkki tällaisesta hierarkisesta rakenteesta (Hovi ym. 2005, 56-58). Taulun muita sarakkeita ovat alku- ja loppusarake sekä tilityyppi, joka kertoo, onko kyseessä pelkkä otsikko vai nimike, jonka kohdalle tulostetaan myös saldo. Alkusarake kertoo, mistä asti kirjanpitotilit kuuluvat kyseiseen tilinpäätöstiliin ja loppusarake sen, mihin asti kirjanpitotilit kuuluvat kyseiseen kirjanpitotiliin. Alku- ja loppusarakkeessa on tarvittavan pitkä merkkijono kirjanpitotilin alusta halutun tarkkuuden saavuttamiseksi.

Esimerkiksi ostot ovat materiaalien ja palvelujen alla, mutta samalla myös liikevoiton alla sekä voiton ennen tilinpäätössiirtoja ja veroja alla. Tarkkaan ottaen ne ovat myös tilikauden voiton alla, mutta sen esittää itse asiassa tuloslaskelman alla olevien kaikkien kirjanpitotilien summa. Osalla kirjanpitotilejä pitäisi siis olla kaksi emotiliä, jolloin puuhun muodostuisi lenkkejä. Tilinpäätös ei siis olekaan puu vaan tarkkaan ottaen verkko. Silmukat voitaisiin hoitaa esimerkiksi omalla tietokantataululla, jossa on erikseen ohjaustilien emotilit ja lisätilit (=puun lenkit). Esimerkiksi liikevaihdolla olisi lisäemotilit liikevoitto ja voitto ennen tilinpäätössiirtoja ja veroja. Tällaista rakennetta kutsutaan BOM-rakenteeksi (bill-of-materials) (Hovi ym. 2005, 58-59).

Voidaan huomata, että alku- ja loppusarakkeita ei itse asiassa tarvittaisi lainkaan, jos hierarkiataulu ja kirjanpitotilien taulu muodostaisivat yhdessä yksikäsitteisen puurakenteen ilman lenkkejä. Kirjassa Algorithms in C on esimerkki puurakentei-

den käymisestä lävitse erilaisissa järjestyksissä (Sedgewick 1998, 232). Jos tietokannan puurakenne purettaisiin ohjelmointikielen puurakenteeksi ja käytäisiin saatu puu läpi esijärjestyksessä (preorder), saataisiin tulostettua tilinpäätös. Kukin tilinpäätöksen nimikkeen (ohjaustilin) arvo (esim. oman pääoman määrä) olisi puussa nimikkeen takana olevien kirjanpitotilien summa. Tässä ratkaisussa tilien numerointi olisi täysin vapaa. Kirjanpitotilien numerointi täysin vapaasti ei liene kuitenkaan tarkoituksenmukaista. Silmukoiden implementointi ohjelmointikielelle ja niiden läpikäynti lisäisi ohjelman monimutkaisuutta. Silloin pitäisi ratkaista edellisessä kappaleessa kuvatun BOM-taulukkorakenteen läpikäynti. Tässä työssä päädyttiin säilyttämään hierarkiataulukon sarakkeet, joissa on tiliohjauksen alkua ja loppunumerot ja käyttämään niiden tietoja tilinpäätöksen muodostamiseen.

Tilinpäätöstä varten kirjanpito-ohjelmassa pitää käydä kirjaustaulussa läpi käytössä olevat tilit ja laskea tilien saldot sekä siirtää ne tilien ohjauksen mukaisesti tuloslaskelmaan ja taseeseen. Jos tilinpäätöksen tiedot tallennetaan omaan tauluunsa, saadaan tilinpäätös nopeasti tietokannasta, mutta silloin syntyy vaara, että tilinpäätös jätetään muodostamatta uudelleen käyttäjän tehtyä muutoksia kirjanpitoon. Jotta tietoja voidaan käyttää seuraavana vuonna vertailutietoina, kannattaa ne kuitenkin tallettaa omaan tauluunsa viimeistään tilikautta suljettaessa. Ohjelmaa suunniteltaessa päätettiin tallentaa tiedot tilinpäätöstauluun aina tilinpäätöstä tehtäessä. Samoin tehdään kirjanpitotilien tilikohtaisille saldoille, jotka tallennetaan omaan summatauluunsa nopeuttamaan tilinpäätöksen muodostamista.

Tässä summataulussa on perusavain käytettyjen kirjanpitotilien numeroille. Se toimii samalla viiteavaimena kirjanpitotilien taulun tilinumerosarakkeeseen. Toinen sarake on kirjanpitotilien tilikohtaiset saldot. Taulua voidaan hyödyntää normaalin taseen ja tuloslaskelman muodostamisen lisäksi myös tilikohtaisen tuloslaskelman ja taseen muodostamisessa aputauluna. Summataululla ja tilinpäätöstaululla ei ole suoraa yhteyttä (Kuva 6). Tilinpäätöstaulun perusavain on kirjanpitotilien tilinumero, joka toimii samalla viiteavaimena hierarkiataulun tilinpäätöstilien tilinumeroon. Tilinpäätöstaulun toinen sarake on tilinpäätöstilin saldo.

Tilinpäätöstaulu

tp_tili_no (va)	saldo
...	...
028	-4356,00
0280	-2754,00
0290	-1602,00
030	-45000,00
...	...

Summataulu

kp_tili_no (va)	saldo
	...
292040	-558,00
3002	-40000,00
4000	327,00
701001	6000,00
	...

Kuva 6 Tilinpäätöstaulun ja summataulun väiltä puuttuu suora yhteys

Liitetiedoille varataan kaksi taulua. Toinen on kaikille käyttäjille yhteinen liitetietojen hierarkian pohja. Käyttäjä ei siis pääse muokkaamaan liitetietojen rakennetta. Liitetietojen hierarkian taulun perusavaimeksi valitaan numero, joka kuvaa PMA-asetuksen 3 luvun liitetietojen pykälänumerointia 13 § asti. Numeroksi 14 valitaan tilinpäätöksen laatimisessa noudatettu säännöstö, numeroksi 15 oman pääoman muutokset sekä hallituksen ehdotus jakokelpoisen vapaan oman pääoman käytöstä, numeroksi 16 tarkistesummat ja numeroksi 17 päiväys ja allekirjoitukset. Asetuksen 3 luku 1 § tilinpäätöksen laatimisesta alakohdat on numeroitu alkaen numerosta 101. Hierarkian muodostamiseksi käytetään emotilisaraketta samoin kuten tuloslaskelman ja taseen yhteisessä hierarkiataulussa. Taulun muina sarakkeina käytetään liitetietojen kohtien esittämisjärjestystä, otsikkosaraketta ja kyseiseen kohtaan liittyvien opasteiden saraketta.

Käyttäjäkohtaisten liitetietojen taulun perusavain, numero, on samalla viiteavain edellä kuvatun yhteisen liitetietohierarkiataulun perusavaimen. Käyttäjäkohtaisen taulun sisältösarakkeeseen pitää varata riittävästi tilaa liitetiedolle ja linkkisarakkeeseen voidaan tallentaa liitetietotositteen osoite. Lisäksi käyttäjäkohtaisessa liitetietotaulussa on sarake, jonka arvo kertoo, näytetäänkö kohta liitetiedoissa vai ei. Eheyssäännöksi viiteavaimelle sopii vain rajoittaminen, sillä isätaulu on kaikille yhteinen, eikä yksi kirjanpitovelvollinen saa vaikuttaa toisiin kirjanpitovelvollisiin. Eheyssäännöt toimivat turvana suunnittelu ja ohjelmointivirheitä vastaan ja ne on hyvä määrittää valmiiksi myöhempiä muutoksia varten. Tässä käyttäjille ei anneta mahdollisuutta muuttaa perus- tai viiteavaimen arvoja liitetietojen käyttöliittymässä.

5.3.4 Muut taulut

Käyttäjän kunkin tilikauden alku- ja loppumisajankohta tallennetaan kaikkien käyttäjien yhteiseen tilikausitauluun. Tästä syystä taulun perusavaimena toimii tilikauden juoksevan numeron lisäksi käyttäjän numero, joka on samalla viiteavain käyttäjätauluun. Muut sarakkeet ovat tilikauden alku- ja loppumisajankohta. Eheyssäntönä käytetään rajoittamista, vaikka tässäkään käyttäjälle ei anneta mahdollisuutta muuttaa perus- ja viiteavaimen arvoja. Tätä taulua voitaisiin käyttää varmistamaan, että kirjaukset ovat tilikauden sisällä.

Rekisteröitymistä varten ohjelmassa on rekisteröintitaulu. Käyttäjälle lähetetyssä rekisteröinnin vahvistussähköpostissa on rekisteröinnin vahvistuslinkki. Sen GET-parametrin arvo toimii perusavaimena rekisteröintitaulussa. Taulun muut sarakkeet ovat sähköpostiosoite tiivisteenä tai sellaisenaan ja tiiviste käyttäjän syöttämästä salasanasta. Rekisteröinnin vahvistamisen yhteydessä rekisteröintitaulun käyttäjätiedot lisätään käyttäjätauluun ja poistetaan rekisteröintitaulusta. Taulujen välillä ei ole viiteavainta.

Käyttäjätaulun perusavaimena toimii juokseva käyttäjänumerointi. Se toimii käyttäjän tunnisteena ohjelmassa ja kytkee käyttäjän oikeisiin tauluihin. Käyttäjän salana ja sähköpostiosoite, jotka siirretään rekisteröintitaulusta käyttäjätauluun, tarkistetaan aina kirjautumisen yhteydessä. Käyttäjätauluun tallennetaan rekisteröinnin vahvistamisen IP-osoite ja viimeisen kirjautumisen IP-osoite sekä molempien ajat. Taulussa on vielä lisäksi sarake nykyiselle tilikauden numerolle, suurimmalle käytössä olevalle tilikauden numerolle ja suurimmalle sallitulle tilikauden numerolle, jonka jälkeen tilikausia ei enää säilytetä tietokannassa.

Jos haluttaisiin toteuttaa kustannuspaikat, ne tallennettaisiin kustannuspaikkojen tauluun, jonka perusavain olisi kustannuspaikkanumero ja toinen sarake olisi kustannuspaikan nimi. Kustannuspaikkanumerosarake pitäisi lisätä kirjaustauluun vierasavaimena. Vaihtoehtoisesti se voitaisiin lisätä omaan tauluunsa vierasavaimena. Tauluun tulisi perusavaimeksi ja samalla vierasavaimeksi kirjausnumero kirjaustaulusta ja perusavaimeksi lisäksi tositenumero ja tositelaji joko kirjaus- tai tositetaulusta. Varsinkin jälkimmäisen vaihtoehdon toteutus myöhemmin olisi helppoa. Se ei muuttaisi toiminnassa olevan tietokannan tauluja. Erikseen

pitäisi lisätä syöttölomakkeelle kohta kustannuspaikka ja toteuttaa toiminnallisuus, jossa käytäisiin haluttuja asioita lävitse kustannuspaikoittain raportoinnissa. Kustannuspaikkamahdollisuutta ei lisätä asiakasvaatimuksiin, vaan se käsiteltiin tässä vain esimerkkinä siitä, miten hyvin suunniteltuun tietokantaan on helppo lisätä tauluja ja ominaisuuksia.

Tietokantojen indeksointien virittämistä ei suunnitella, sillä toimiva kirjanpitolain mukainen ohjelma on riittävä vaatimus eikä hakujen suoritus aika ollut vaatimuksissa.

5.4 Ohjelman toteutus ja testaus

Koska työ on toiminnallisen opinnäytetyön kaltainen, pidettiin ohjelman tekemisestä päiväkirjaa, jota käytettiin opinnäytetyön kirjoittamisessa apuna (Vilka & Airaksinen 2003, 19-22). Ohjelman kehitys toteutettiin kannettavalla tietokoneella LAMP-alustalla (Linux, Apache-web-palvelin, PHP ja MySQL-tietokanta) paikallisella palvelimella. Kehityksessä käytettiin git-versionhallintajärjestelmää, joka ohjelmakoodin kommenttien ja päiväkirjan lisäksi toimii työn toteutusvaiheen dokumentaationa. Teknisiin ongelmiin löytyi hyvin ratkaisuja keskustelupalstoilta.

Kirjanpito-ohjelman projektinhallinta oli melko suoraviivaista, koska toteuttajia oli vain yksi. Projektin pienen koon takia ei käytetty mitään luvussa 4 mainittua mallia suoraan. Menetelmä oli lähempänä ketteriä menetelmiä kuin vesiputousmallia. Tietokantaa kehitettiin ja testattiin käytännössä osittain samaan aikaan käyttöliittymän kanssa. Ohjelmaa kehitettiin prototyypinä, johon lisättiin uusia ominaisuuksia tekemällä aina uusi verkkosivu uudelle tarvittavalle tehtävälle. Käytännössä tietokantasuunnittelu olisi kannattanut kuitenkin suorittaa loppuun asti ennen kuin verkkosivuja oli tehty liikaa, sillä myöhemmin havaitut puutteet suunnittelussa lisäsivät tarvetta tehdä muutoksia ohjelmakoodiin jälkikäteen. Osa toteutetuista ratkaisuista olisi johtanut vaatimusmäärittelyn muutoksiin jo suunnitteluvaiheessa, jos suunnittelu olisi tehty kokonaan loppuun asti ennen työn aloittamista. Luvussa 5.4.1 kerrotaan aluksi, minkälaisia muutoksia luvussa 5.3 kuvattujen taulujen rakenteisiin päätettiin tehdä toteutuksen aikana.

5.4.1 Suunnitellusta poikkeava toteutus

Sivuille, joissa käsiteltiin kirjauksia, toteutettiin käyttäjälle mahdollisuus valita kirjanpitotili valikoista. Valikoissa näytetään kirjanpitotilin nimi ja numero. Valittavien tilien määrä vähenee sitä mukaa, kun käyttäjä kirjoittaa tilinumeron numeroita tai tilin nimeä. Etsintä tapahtuu mistä tahansa sanan osasta, ei vain alusta alkaen. Valinta toteutettiin datalist-elementeillä. Eri selaimet näyttävät valikon eri tavoin ja toteuttavat haun valikosta eri tavoin. Esimerkiksi Internet Explorer ei näyttänyt valikoita ollenkaan ja Chromium-selain näytti valikon vain ensimmäisessä data-elementissä. Edge-, Chrome- ja Firefox-selaimet näyttivät valikon oikein. Datalist-elementti toteutettiin JavaScriptillä. JavaScript-tiedosto sijoitettiin siirtokustannusten minimoimiseksi muualle kuin palvelimelle, josta muu sivun koodi jaeltiin.

Tällaista erillistä JavaScript-tiedostoa ja sen sisältämää valikkoa ei voi muuttaa yhtä helposti kuin palvelimelta jaettavaa valikkoa. Ongelman kiertämiseksi tietokantaan perustettiin aluksi yksi ”pääkäyttäjä”, jonka käytössä oleviin alkuperäisiin kirjanpitotileihin ei sallittu tehdä muutoksia. Alkuperäiset kirjanpitotilit olivat samoja JavaScript-tiedostossa olevien kanssa. Kun uusi käyttäjä halusi lisätä omia kirjanpitotilejä, lisättiin ne yhteiseen kirjanpitotilien tauluun, sillä lapsitaulun (kirjaustaulu) viiteavaimella tilinnumero voi viitata vain yhteen isätauluun (kirjanpitotilien taulu). Tällainen menettely edellytti kuitenkin käyttäjä-ID-sarakkeen lisäämistä kirjanpitotilien taulun perusavaimen, jotta kirjanpitotilien taulun perusavain säilyy yksilöllisenä ja jotta käyttäjien tilit voidaan pitää erillään toisistaan. Tämä taas edellytti käyttäjä-ID-sarakkeen lisäämistä myös lapsitauluun kirjaus, jotta voidaan erottaa käyttäjän omat lisäykset ”pääkäyttäjän” alkuperäisistä tileistä. Menettely mahdollistaa sen, että muut käyttäjät pystyvät uudelleennimeämään ja -ohjaamaan pääkäyttäjän käytössä olevia tilinumeroita. Ohjelman toteutettiin niin, että siinä käytetään aina käyttäjän omaa tiliä, mikäli sillä on sama tilinnumero kuin ”pääkäyttäjän” tilillä.

Myös summat tauluun, jonka isätaulu on käytettyjen kirjanpitotilien taulu, olisi pitänyt lisätä käyttäjä-ID-sarake. Toteutuksessa tässä tapahtui kuitenkin virhe ja käyttäjä-ID-sarake jäi pois summatauluista. Liitetietojen hierarkian ja liitetietojen taulussa käytettiin turhaan myös emonumeroa perusavaimena, sillä jo luvussa 5.3 esitetty asetuksen numerokohtiin perustuva avain on yksikäsitteinen.

Tietojen lisäys ja muokkaus, kirjanpitotilien käsittely, pää- ja päiväkirja sekä tilinpäätöksen toteutus ohjelmoitiin ennen uuden tilikauden avaamisen, vanhan tilikauden sulkemisen ja tilikauden vaihdon suunnittelua. Koska ohjelmakoodia ei haluttu muuttaa jälkikäteen, päädyttiin säilyttämään nykyisen tilikauden taulujen nimet t samoina käsiteltävästä tilikaudesta riippumatta. Tähän päästiin kopiaamalla tilikauden vaihdon ($t_0 \rightarrow t_1$) yhteydessä ensin nykyiseen tilikauteen t tehdyt muutokset tilikauden t_0 tauluiksi. Sen jälkeen kopioidaan sen tilikauden taulut t_1 , johon ollaan siirtymässä nykyisen tilikauden tauluiksi t . Näin muutokset tehdään aina vain yhden nimisiin tauluihin t . Tämä edellyttää aina ennen kopiointia sen varmistamista, että tilikausi t_n johon kopioidaan, ei ole lukittu.

Tilikauden t_1 sulkemisessa päädyttiin kopioimaan se tilikaudeksi t_2 . Vastaavasti aikaisemmat tilikaudet kopioidaan samalla tavalla, jos niitä on. Koska tietoja ei haluta menettää, aloitetaan kopiointi vanhimmasta tilikaudesta alkaen. Näin tilikauden taulut on nimetty aina uusimmasta tilikaudesta t_0 vanhimpaan tilikauteen t_n . Tilikauden avaamisessa ei samanlaista ongelmaa syntynyt, sillä käyttäjää vaaditaan sulkemaan tilikausi t_1 ennen tilikauden t_0 avaamista. Silloin voidaan vain kopioida tilikausi t_0 tilikaudeksi t_1 ja avata uusi tilikausi t_0 .

5.4.2 Muu toteutus ja testaus

Tässä esitetään esimerkkinä tilinpäätösfunktion toteutus. Ensin haetaan aputauluun (summataulu) kirjanpitotilien tilinumerot ja saldot. Sitten käydään aputaulu lävitse seitsemän kertaa silmukassa. Silmukassa indeksin arvo käy läpi arvot kolmesta yhdeksään. Indeksien arvot ovat kirjanpitotilien etunumeroita. Jokaisella kierroksella lisätään tuloslaskuriin niiden kirjanpitotilien saldot, jotka alkavat samalla numerolla kuin indeksin arvo on sillä kierroksella. Tämä riittää koska ollaan kiinnostuneita vain tilikauden tuloksesta eikä vielä tilinpäätöksestä. Silmukka käy siis lävitse kaikki tuloslaskelman tilit ja laskee niiden yhteissumman eli tilikauden tuloksen tuloslaskuriin. Tilikauden tulos lisätään aputauluun (summataulu) tilikauden tulokseksi oman pääoman alla olevalle kirjanpitotilille, jotta oma pääoma ja tilikauden tulos saadaan menemään oikein taseessa.

Käydään seuraavaksi uudessa silmukassa ohjelmallisesti lävitse tilinpäätöksen hierarkia. Kutakin tilinpäätöshierarkian riviä kohti lasketaan aputaulusta (summa-
taulu) yhteen niiden kirjanpitolien saldot, joiden tilinumeron alku on tilinpäätöshierarkian alku- ja loppusarakkeen välissä. Saatu arvo ja sitä vastaava tilinpäätöstilin tilinumero viedään tilinpäätöstauluun. Kun tuloslaskelma sitten haluttiin tulostaa ohjelmassa näytölle, kutsuttiin ensin tilinpäätösfunktiota. Sen jälkeen toteutettiin tilinpäätöstaulun ja tilinpäätöksen liitoksen läpikäynti, jonka läpikäynnin tulos näytetään käyttäjälle tilinpäätöksenä.

Tietokantakyselyissä käytettiin PHP:n PDO-tietokanta-abstraktiokerrosta. Silloin ohjelman alla olevaa tietokantaa voidaan vaihtaa pienellä työllä ja samalla saavutetaan helpommin parempi tietoturvan taso. Tietokantana käytettiin MariaDB-tietokantaa, joka on MySQL-tietokannan kanssa yhteensopiva. Taulujen luonnit, lisäykset tauluihin sekä poistot ja haut tauluista yritettiin pitää mahdollisimman hyvin yhteensopivina muihin tietokannan hallintajärjestelmiin verrattuna, jotta siirrettävyys pysyisi mahdollisimman hyvänä. Viennit pyrittiin peruttamaan (rollback) niiden epäonnistuessa. Tietokannan samanaikaisuuden hallintaa ei toteutettu, koska oletusarvoisesti kirjauksia yhdelle kirjanpitovelvolliselle tekee vain yksi käyttäjä kerrallaan. Pilvessä toimivassa kirjanpito-ohjelmassa vientejä samalle kirjanpitovelvolliselle voisi tosin helposti tehdä monta käyttäjää yhtä aikaa.

Ohjelma pyrittiin toteuttamaan mahdollisimman modulaarisena ja yleisenä. Esimerkiksi taulujen luomiseksi kehitettiin oma luokkansa PHP-kielellä. Siten kaikki tietokannan taulujen luonnit ovat samanlaisia kaikkialla ja mahdollinen muutos taulujen luonnissa pitää tehdä vain yhdessä paikassa. Kaikki tietokantaan liittyvät nimet tauluista sarakkeisiin koodattiin PHP-muuttujien taakse, niin että ne ovat samoja kaikkialla ohjelmassa. Siten tietokannan muuttujanimien vaihto voidaan tehdä muuttamalla arvoja yhdessä tiedostossa. Toistuvat operaatiot pyrittiin sijoittamaan funktiokutsujen taakse. Toistuvia toimintoja myös muutettiin myöhemmin funktioiksi. Esimerkiksi HTML-sivun alun ja lopun tulostus muutettiin PHP-funktioiksi, jolloin ne ovat kaikkialla yhtenevät. Sivujen yhteinen navigointivalikko pidettiin yhdessä tiedostossa. Sivujen ulkoasun hallinnointi toteutettiin CSS:n avulla (Cascading Style Sheets). SQL-lauseet, joilla lisättiin tilikartan tilit kirjanpi-

totilien tauluun, tilinpäätöstilit tilinpäätöstilien hierarkiatauluun ja liitetietojen alkutiedot liitetietoja hierarkiatauluun ja liitetietotauluun ylläpidettiin ja tuotettiin tekstinkäsittelyohjelmalla.

Käyttäjien pääsynhallinta oikeisiin tietoihin toteutettiin kirjauksen yhteydessä käyttäjään sidotulla palvelinpuolen istuntotunnuksella. Kirjanpitovelvolliselle palvelun rekisteröitymisen vahvistuksen yhteydessä annettua ID-numeroa käytettiin istuntotunnisteena ja taulujen nimissä oikeuttamaan käyttäjän pääsy omiin tauluihinsa.

Tilikartan tilit numeroitiin raportointikoodiston pohjalta, jotta mahdollinen myöhempi integrointi XBRL-raportointiin olisi helppo tehdä. Raportointikoodiston konsernitilejä ja muita mikroyritykselle tarpeettomia tilejä poistettiin tilikartasta. Myös lisäys- ja vähennystilit jätettiin pois. Toisaalta tilikarttaan lisättiin henkilöyhtiöiden oman pääoman tilejä, sillä niitä on raportointikoodistossa melko vähän. Myös työntekijän sosiaaliturvamaksuja eriteltiin raportointikoodistoa tarkemmin. Jos ohjelma olisi toteutettu aatteelliselle yhdistykselle, olisi tilejä pitänyt lisätä paljon, sillä aatteellisen yhdistyksen tilit liiketoimintaa lukuun ottamatta puuttuvat raportointikoodistosta täysin. (Tili-instituuttisäätiö 2017.)

Testausta tapahtui koko ohjelman kehityksen ajan. Ohjelman kaikki kohdat, joissa käyttäjä sai syöttää tietoja, käytiin läpi ja tarkastettiin, että käyttäjäsyötteet on sidottu oikein PDO:n avulla. Sen jälkeen ohjelmaa testattiin vielä SQL-injektioiden varalta. Lopuksi ohjelma testattiin HTML-validaattorilla ja puutteita korjattiin. Cross site scripting (XSS) haavoittuvuuksia ei testattu. Periaatteessa kenenkään toisen käyttäjän ei pitäisi päästä lisäämään koodia alueelle, joka on toisen käyttäjän nähtävissä, mutta käyttäjäsyötteestä poistettiin silti erikoismerkit XSS-haavoittuvuuksien vähentämiseksi.

Tutkimuskysymyksen 2 vastauksen keskeisimmät kohdat on esitetty vielä taulukossa 2. Ohjelmaa voi testata osoitteessa <http://www.palvelen.fi/kipi/> toistaiseksi. Työn tekijä ei ota maksua ohjelman käytöstä eikä myöskään vastuuta ohjelman käytöstä aiheutuneista vahingoista.

1 Ensin tehtiin vaatimusmäärittely
<p>Kirjanpidon sääntelyn osalta vaatimukset on esitetty taulukossa 1. Tässä esitetään opinnäytetyön tekijän asettamia vaatimuksia.</p> <ul style="list-style-type: none"> - Ohjelmassa pitää olla mahdollisuus käyttää kirjauksissa selitteitä sekä tosienumeron lisäksi tositelajia ja tallentaa linkki tositteeseen. - Käyttäjän pitää saada muokata kirjanpitotilejä, mutta ei tilinpäätöstilejä eikä tilinpäätöstä. - Ohjelmassa pitää olla mahdollisuus korjata virheitä. - Taulukossa 1 sulkujen sisällä olevia vaatimuksia ei toteuteta ohjelmassa eikä kaikkia tuloslaskelmakaavoja. - Ohjelman on oltava otettavissa kirjanpitovelvolliselle helposti käyttöön internetissä. - Ohjelma suunnitellaan jaettuun webhotelliin ja sen vaatima tietoliikenteen määrä ja tietojen vaatima säilytystila on pidettävä pienenä.
2 Suunnittelu seurasi vaatimusmäärittelyä
<p>Suunnittelussa tietokantasuunnittelulla oli suuri osuus. Ensin muodostettiin käsiteanalyysillä käsitelmä, joka normalisoitiin suoraan tietokannan tauluiksi. Ratkaisuperiaatteen suunnittelulla oli pienempi osuus. Siinä pohdittiin toteutustapaa ja toteutuksen teknisiä ratkaisuja.</p>
3 Toteutus alkoi jo osittain tietokantasuunnittelun kanssa samaan aikaan
<p>Ohjelmasta oli jo varhaisessa vaiheessa prototyyppi, jota kehitettiin samaan aikaan tietokantasuunnittelun kanssa lisäämällä aina uusia sivuja uusia ominaisuuksia varten. Modulaarisuutta kehitettiin tekemällä toistuvista asioista moduuleita. Koska tietokantasuunnittelua ei tehty loppuun asti ennen ohjelman toteutuksen aloitusta, tehtiin suunnittelussa ratkaisuja, joita ei olisi tehty jos suunnittelu olisi viety loppuun asti ennen toteutuksen aloitusta. Osittain toteutuksen yhteydessä tehdyt havainnot muuttivat tietokantasuunnittelua, mutta se ei ulottunut vaatimusmäärittelyihin asti, niin että vaatimuksia olisi muutettu sopivimmiksi.</p>
4 Testausta tehtiin koko toteutuksen ajan
<p>Vasta lopussa tehtiin suunnitelmallinen testaus. Ohjelman aikana tehty testaus oli lähinnä ohjelman kehityksen aikana tapahtunutta toimintojen koe-käyttöä. Virhelogeja ja keskustelupalstoja hyödynnettiin sekä toteutuksessa että testauksessa</p>

Taulukko 2 Tiivistelmä tutkimuskysymyksen 2 vastauksista

6 Arvio työstä ja jatkokehitys

Työn alussa selvitettiin, millaisia vaatimuksia lainsäädäntö ja hyvä kirjanpito-asettavat kirjanpito-ohjelmalle. Se vastasi ensimmäiseen tutkimuskysymykseen: Millaisia vaatimuksia ja rajoitteita lainsäädäntö ja hyvä kirjanpito-asettavat mikroyritykselle soveltuvalla kirjanpito-ohjelmalle? Vaatimukset käydään lävitse yksityiskohtaisesti luvussa 2. Ne on vielä koottu taulukkoon 1. Samalla vastaus ensimmäiseen tutkimuskysymykseen yhdessä opinnäytetyön tekijän asettamien lisävaatimusten kanssa toimi lähtökohtana toiseen tutkimuskysymykseen vastaamiseksi.

Miten vaatimukset ja rajoitteet täyttävä kirjanpito-ohjelma suunnitellaan selain-pohjaisena ja miten se toteutetaan? Myös tähän toiseen tutkimuskysymykseen saatiin vastaus: pystyttiin suunnittelemaan ja toteuttamaan vaatimusten mukainen kirjanpito-ohjelma. Työn keskeiset vaiheet on koottu taulukkoon 2 yhteenvedoksi. Työ vahvisti ohjelmistotekniikan vakiintunutta käsitystä, että mitä aiemmassa vaiheessa muutokset tehdään, sitä pienemmällä työmäärällä ne pystytään tekemään. Työn merkittävin suunnittelutieteellinen tulos oli se, että suunnittelu pitää aina viedä loppuun asti ja tarkastella vaatimusmäärittelyjen hyväksyttävyyttä uudestaan suhteessa suunnittelun lopputulokseen ennen toteutuksen viemistä liian pitkälle. Näin toimien voidaan tarvittaessa muuttaa vaatimusmäärittelyä heti alussa, jotta toteutukseen ei tarvitsisi tehdä muutoksia jälkikäteen.

Jos tietokantasuunnittelu olisi tehty loppuun asti ennen toteutuksen aloitusta, olisi todennäköisesti luovuttu vaatimuksesta minimoida tiedonsiirto tilikartan siirtämisen osalta. Silloin tilikarttaa ei olisi tallennettu erilliseen JavaScript-tiedostoon ja tilit olisivat tositteita syötettäessä olleet käytössä ilman JavaScriptiä. Ennen kaikkea kaikille käyttäjille olisi kuitenkin silloin voitu tehdä käyttäjäkohtainen kirjanpitolien taulu, jos myös vaatimuksesta minimoida tallennettavan tiedon määrä olisi luovuttu. Silloin kaikilla käyttäjillä olisi ollut samat aloitustilit, joita käyttäjät olisivat saaneet muokata mielensä mukaan eikä käyttäjä-ID-sarakkeita olisi tarvittu. Jos kaikkien kirjanpitolien taulusta (tilikartta) olisi tehty tilikausikohtainen, olisi mahdollista poistaa edellisenä tilikausina käytössä olleita kirjanpitolilejä. Samaan päästäisiin myös vaihtamalla kirjaukset viittaamaan käytettyjen tilien vuosikohtaisiin tauluihin. Jälkimmäinen säästäisi tilaa, mutta se olisi työläämpää.

Taulujen nimeäminen pitäisi toteuttaa uudelleen niin, että tauluja ei enää tarvitsisi kopioida tilikauden vaihteessa ennen kuin maksimaalinen säilytettävien taulujen määrä on saavutettu. Vanhimman tilikauden taulut voisivat olla nimetty tauluiksi t_0 ja uusimman tilikauden taulut tauluiksi t_n . Silloin seuraavan tilikauden taulut nimettäisiin tauluiksi t_{n+1} . Vaihtoehtoisesti tilikauden siirto taulujen sarakkeeksi poistaisi koko nimeämisiongelman. Molemmat ratkaisut vaatisivat muutoksia ohjelmakoodiin monessa paikassa.

Jos tuloslaskelman ja taseen hierarkiataulu ja liitetietojen hierarkiataulu muutettaisiin käyttäjäkohtaisiksi, voitaisiin käyttäjien antaa muuttaa tilinpäätöksen rakennetta. Näin toimien käyttäjä voisi itse muokata yrityksen tilinpäätöksestä esimerkiksi yhdistyksen tilinpäätöksen. Jotta tilinpäätös olisi oikea tilinpäätöksen muutoksen jälkeen eri tilikausien välillä, pitäisi tuloslaskelman ja taseen hierarkiataulu ja liitetietojen hierarkiataulu muuttaa myös tilikausikohtaiseksi. Näin toimimalla kaikki muut taulut paitsi tilikausien taulu sekä käyttäjä- ja rekisteröitymistaulu olisivat lopulta käyttäjä- ja tilikausikohtaisia.

Edelliselle vastakkainen varteenotettava vaihtoehto jatkokehityksen kannalta olisi siirtyä kaikille käyttäjille yhteisiin tauluihin, joissa käyttäjännumero olisi sarakkeena kaikissa kirjanpitoon liittyvissä tauluissa. Ainakin tilikausien osalta olisi harvittava siirtymistä käyttämään tilikautta taulujen sarakkeena tilikausikohtaisten taulujen sijaan, kuten Pupesoft-toiminnanohjausjärjestelmässä on tehty. Tilikausien osalta valittuihin ratkaisuihin päädyttiin pitkälti siitä syystä, että ohjelman toteutusta tehtiin tilikauden vaihtoon asti ennen kuin alettiin suunnitella tilikauden vaihdosta. Käyttäjien osalta olisi ollut teknisesti mahdollista ottaa käyttöön kaksi käyttäjää testaukseen aikaisemmin. Silloin olisi jouduttu ottamaan aikaisemmin kantaa taulurakenteeseen käyttäjien osalta. Ennen taulurakenteiden muuttamista mihinkään suuntaan on kuitenkin syytä analysoida muutoksen seurauksia ja tutkia muita olemassa olevia ratkaisuja tarkemmin.

Tilinpäätöksen esittäminen pitäisi taseen osalta muuttaa vastaamaan KILA:n suositusta, jossa vastaavan ja vastattavan määrät esitetäisiin alhaalla niiden jälkeen eikä otsikon perässä kuten nyt toteutuksessa tehtiin. Tämä voitaisiin tehdä helpommin, jos tilinpäätöshierarkia luettaisiin puurakenteeseen, kuten luvussa 5.3.3 kuvattiin. Jotta käyttäjä voisi itse muokata tilinpäätöksen hierarkiaa, pitäisi

käyttöliittymään lisätä uusia sivuja. Käyttäjän voitaisiin antaa valita asetuksen mukaiset muutkin kaavat kuin liiketoimintaa harjoittavan yrityksen kaavat, esimerkiksi yhdistyksen ja asunto-osakeyhtiön kaavat. Samalla käyttäjälle voisi tarjota valmiita tilikarttoja, jotka liittyvät enemmän kuhunkin yhtiömuotoon.

Ohjelman jatkokehitys kannattaisi aloittaa edellä mainituista muutoksista ja parannuksista. Vasta sen jälkeen ohjelmaan kannattaisi lisätä uusia ominaisuuksia. Tällainen ominaisuus voisi olla XBRL-raportointi verotusta ja kaupparekisteriä varten. Merkittävä ominaisuus olisi alv-automatiikka, mutta sen ajanmukaisena pitäminen vaatii kohtuullisesti työtä kuten XBRL-raportoinninkin. Jos ohjelmaa lähdetäisiin kehittämään toiminnanohjausjärjestelmän suuntaan, kannattaisi siihen kehittää ensimmäisenä laskutusohjelma ja myyntireskontra ja vasta sen jälkeen ostoreskontra. Tässä oletuksena on, että ohjelmaa kehitettäisiin palveluyritykselle eikä tavaroita myyvälle mikroyritykselle.

Kuvat

Kuva 1 Vientilausekkeita tiliristikoina esitettynä	20
Kuva 2 ER-kaavio tositteiden ja kirjauksen välillä. Nimen lopussa oleva (va) tarkoittaa vierasavainta	24
Kuva 3 Kuvan 2 mukaisen ER-kaavion esitys tauluina UML:n mukaisin nuolin	26
Kuva 4 Päiväkirjanpitotaulu ennen normalisointia	37
Kuva 5 Tilinpäätöksen hierarkiataulun, kirjanpitotilien taulun ja kirjaustaulun väliset suhteet	39
Kuva 6 Tilinpäätöstaulun ja summataulun väliltä puuttuu suora yhteys	42

Taulukot

Taulukko 1 Tiivistelmä tutkimuskysymyksen 1 vastauksista	29
Taulukko 2 Tiivistelmä tutkimuskysymyksen 2 vastauksista	49

Lähteet

Devlab 2017a. Toiminnanohjausjärjestelmä pk-yritysten tarpeisiin. <http://www.devlab.fi/>. Luettu 31.11.2017.

Devlab 2017b. Kirjanpito. <http://www.devlab.fi/pupesoft/kayttoohjeet/taloushallinto/kirjanpito/>. Luettu 31.11.2017.

Haikala, I. & Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. Helsinki: Talentum.

Haikala, I. & Märijärvi, J. 2006. Ohjelmistotuotanto. Helsinki: Talentum.

HE 89/2015. Hallituksen esitys eduskunnalle laiksi kirjanpitolain ja eräiden siihen liittyvien lakien muuttamisesta sekä puunkorjuuta aarniometsissä harjoittavien yritysten viranomaisille suorittamien maksujen julkistamista koskevaksi laiksi.

Helineva, T. 2011. Kirjanpito-ohjelman toteuttaminen. Metropolia ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäytetyö. <https://www.theseus.fi/bitstream/handle/10024/32455/kirjanpito-ohjelman-toteuttaminen.pdf?sequence=1>. Luettu 2.11.2017.

Henkilötietolaki 22.4.1999/523.

Hovi, A., Huotari, J. & Lahdenmäki T. 2005. Tietokantojen suunnittelu & indeksointi. Jyväskylä: Docendo.

Iivari, J. & Venable, J.R. 2009. Action research and design science research - Seemingly similar but decisively dissimilar. ECIS 2009 Proceedings. Paper 73. <http://aisel.aisnet.org/ecis2009/73>. Luettu 2.6.2016

Järvinen, P. & Järvinen A. 2011. Tutkimustyön metodeista. Tampere: Opinpajan kirja.

Kirjanpitolaki 30.12.1997/1336.

Kirjanpitolautakunta 2006. Tuloslaskelman ja taseen esittäminen. [http://ktm.elinar.fi/ktm/fin/kirjanpi.nsf/0/FF99DD42509C1481C225723A00555B47/\\$FILE/TP-YLEIS2006.doc](http://ktm.elinar.fi/ktm/fin/kirjanpi.nsf/0/FF99DD42509C1481C225723A00555B47/$FILE/TP-YLEIS2006.doc). Luettu 4.10.2017.

Kirjanpitolautakunta 2011. Kirjanpitolautakunnan yleisohje kirjanpidon menetelmistä ja aineistoista 1.2.2011. [http://ktm.elinar.fi/ktm/fin/kirjanpi.nsf/0/6E7DF338F7AC91C4C22578550029471E/\\$FILE/KILA%20menetelm%C3%A4ohje%2001%2002%2011\).pdf](http://ktm.elinar.fi/ktm/fin/kirjanpi.nsf/0/6E7DF338F7AC91C4C22578550029471E/$FILE/KILA%20menetelm%C3%A4ohje%2001%2002%2011).pdf). Luettu 4.6.2016.

Marin, A. 2014. Kykylaakso -oppimisympäristön jatkokehitys. Tampereen ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäytetyö.

<http://docplayer.fi/50994559-Kykylaakso-oppimisympariston-jatkoke-hitys.html>.
Luettu 2.11.2017.

Odoo s.a., 2017. The Accounting behind Odoo.
https://www.odoo.com/documentation/user/9.0/accounting/overview/main_concepts/in_odoo.html. Luettu 31.10.2017.

Pohjonen, R. 2002. Tietojärjestelmien kehittäminen. Jyväskylä: Docendo.

Sedgewick, R. 1998. Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching. Addison-Wesley.

Sharma, A. & Sharangdevot, S.S. 2010. Investigating the application of AOP methodology in development of Financial Accounting Software using Eclipse-AJDT Environment. Teoksessa Patel, R.B. & Singh, B.P. (toim.) International Conference on Methods and Models in Science and Technology (ICM2ST-10). American Institute of Physics. AIP Conference Proceedings 1324(1), 224-228.
<http://aip.scitation.org/doi/pdf/10.1063/1.3526200>. Luettu 7.11.2017.

Tili-instituuttisäätiö 2017. Raportointikoodisto vuodelle 2016.
https://www.raportointikoodisto.fi/@Banner/Goto?num=57943&link=/%40Bin/60342/RapKo2016_2JULK26112015PAIV01032016.xlsx. Luettu 7.11.2017.

Toikko, T. & Rantanen, T. 2009. Tutkimuksellinen kehittämistoiminta – Näkökulmia kehittämisprosessiin, osallistamiseen ja tiedontuotantoon. Tampere: Tampere University Press.

Tomperi, S. 2016. Käytännön kirjanpito. Helsinki: Edita.

Valtioneuvoston asetus pien- ja mikroyrityksen tilinpäätöksessä esitettävistä tiedoista 1753/2015.

Vilka, H. 2016. Re: Toiminnallinen tutkimus (opinnäytetyö).
hanna.vilka@me.com. 19.5.2016.

Vilka, H. & Airaksinen T. 2003. Toiminnallinen opinnäytetyö. Helsinki: Tammi.

Liite 1 Käyttötapaukset

1 Rekisteröityminen ohjelman käyttäjäksi

Alkutilanne: Käyttäjä menee ohjelman etusivulle ja painaa rekisteröintilinkkiä.

Kuvaus: Käyttäjä ohjataan sivulle, jossa kysytään hänen sähköpostiosoitettaan ja ohjelman salasanaa sekä pyydetään antamaan kysytyn laskun vastaus. Käyttäjä saa valita haluaako myös sähköpostiosoitteensa salattavan vai esiintyvän tietokannassa salaamattomana. Kun käyttäjä on täyttänyt kaikki kentät ja painanut rekisteröidy-painiketta, häntä pyydetään tarkistamaan sähköpostinsa, mikäli kaikki sujuu hyvin. Käyttäjän sähköpostiin lähetetään linkki, josta painamalla rekisteröinti viedään loppuun.

Lopputilanne: Käyttäjän tiedot on tallennettu tietokantaan ja hänelle on avattu tarvittavat taulut.

2 Kirjautuminen ohjelmaan

Alkutilanne: Käyttäjä on rekisteröitynyt ja menee ohjelman etusivulle.

Kuvaus: Käyttäjä antaa sähköpostiositteen ja salasanan omiin kenttiinsä ja painaa kirjaudu-painiketta. Mikäli kirjaudutaan ensimmäistä kertaa ohjelmaan, käyttäjä ohjataan sivulle, jossa kysytään käytetyn tilikauden alku ja loppupäivämäärä. Onnistuneen kirjautumisen ja tietojen antamisen jälkeen käyttäjälle näytetään valikko ja toivotetaan hänet tervetulleeksi. Samoin tehdään aina onnistuneen kirjautumisen jälkeen. Käyttäjälle näytetään myös tieto viimeisen kirjautumisen ajasta ja IP-osoite, josta kirjautuminen tehtiin.

Lopputilanne: Käyttäjä on kirjautunut sisään ja voi alkaa käyttämään ohjelmaa valikosta.

3 Tositteiden lisääminen

Alkutilanne: Käyttäjä on kirjautunut ja painaa linkkiä lisää tositteita.

Kuvaus: Käyttäjä ohjataan sivulle, jossa hän voi antaa tosittelajin, tositenumeron, päivän, tosittien selitteen ja linkin tositteseen. Sivulla on monta identtistä kirjausriviä, joihin käyttäjä voi syöttää tilinumeron, rahamäärän ja riviselitteen. Tilinumero kentässä käyttäjää saa valita pudotusvalikosta olemassa olevista ja lisäämistään tileistä tilinumeron perusteella haluamansa. Ohjelma ehdottaa edellisen tosittien päiväystä ja tililajin seuraavaa vapaata tositenumeroa. Sivulla on elementti, josta käyttäjä voi lisätä uuden kirjausrivin. Lopuksi käyttäjä painaa painiketta, jolla tiedot viedään tietokantaan.

Lopputilanne: Käyttäjän antamat tiedot on tallennettu tietokantaan sivulla, jolle käyttäjä ohjattiin napin painamisen jälkeen. Onnistuneen kirjauksen jälkeen käyttäjä palautetaan takaisin tosittien lisäämissivulle.

4 Tosittien poistaminen

Alkutilanne: Käyttäjä on kirjautunut ja painaa linkkiä poista tosittia.

Kuvaus: Käyttäjä ohjataan sivulle, jossa hän voi antaa tosittelajin ja tositenumeron ja painaa poista-painiketta

Lopputilanne: Tietokannasta on poistettu annettu tosittien tosittaulusta ja tositteseen liittyneet kirjat kirjaustaulusta.

5 Tosittien muokkaaminen

Alkutilanne: Käyttäjä on kirjautunut ja tarkastelee päiväkirjaa. Hän painaa linkkiä, joka johtaa muokattavaan tositteseen.

Kuvaus: Käyttäjä ohjataan sivulle, jossa hän voi muuttaa tosittelajin, tositenumeron, päivän, tosittien selitteen ja linkin tositteseen. Sivulla on monta identtistä kirjausriviä, joihin käyttäjä voi vaihtaa tai lisätä tilinumeron, rahamäärän ja riviselitteen. Kaikki kenttien tiedot on haettu tietokannasta. Sivulla on elementti, josta käyttäjä voi lisätä uuden kirjausrivin. Lopuksi käyttäjä painaa painiketta, jolla tiedot viedään tietokantaan. Ennen vientiä vanhat tiedot poistetaan. Koska tosittelajia ja tositenumeroa saa muuttaa, sivua voi käyttää myös tietojen kopiointiin olemassa olevan tosittien päälle.

Lopputilanne: Käyttäjän antamat tiedot on päivitetty tietokantaan. Onnistuneen päivityksen jälkeen käyttäjä palautetaan takaisin päiväkirjaan.

6 Oman kirjanpitotilin lisääminen

Alkutilanne: Käyttäjä on kirjautunut ja painaa linkkiä lisää kirjanpitotilejä

Kuvaus: Käyttäjä ohjataan sivulle, jossa hän voi valita linkistä, minkä tilinpäätöstilin alle kirjanpitotili lisätään. Valittuaan linkin, jonka alle kirjanpitotili lisätään, käyttäjä ohjataan sivulle, jossa näytetään tilinpäätöstilin alla jo olevat kirjanpitotilit. Käyttäjä saa syöttää tilinumeron kenttään, johon on annettu kirjanpitotilin numeron alku. Toiseen kenttään syötetään kirjanpitotilin nimi. Lopuksi käyttäjä painaa painiketta lisää tili.

Lopputilanne: Käyttäjän antama kirjanpitotili lisätään kirjanpitotilien tauluun ja päivitetään sivu, jolloin käyttäjä näkee lisäyksen.

7 Oman kirjanpitotilin poisto ja muokkaus

Alkutilanne: Käyttäjä on kirjautunut ja painaa linkkiä muokkaa ja poista kirjanpitotilejä.

Kuvaus: Käyttäjä ohjataan sivulle, jolla näytetään kaikki muokattavat ja poistettavat tilit. Käyttäjän tulee painaa muokattavan tai poistettavan tilin edessä olevaa valintanappia. Sivulla on kenttä uutta nimeä varten ja erilliset painikkeet uudelleennimeämistä ja poistamista varten.

Lopputilanne: Käyttäjän painettua jompaakumpaa painiketta suoritetaan poisto tai uudelleennimeäminen valinnan mukaan. Lopuksi sivu päivitetään, jotta käyttäjä näkee muutoksen.

8 Liitetietojen muokkaus

Alkutilanne: Käyttäjä on kirjautunut ja painaa linkkiä liitetiedot.

Kuvaus: Käyttäjä ohjataan sivulle, jolla näytetään linkkejä eri liitetietoihin. Käyttäjän valittua linkin, hänelle näytetään kyseisen liitetiedon otsikko ja kenttä, johon voi kirjoittaa kyseisen kohdan liitetiedon. Jos liitetieto on jo täytetty, näytetään

senhetkinen sisältö. Sivulla on valintaruutu, jolla säädetään näkykö tieto ja sen otsikko liitetiedoissa. Lisäksi sivulla on kenttä liitetietotositteen internetosoitteelle ja linkkejä muihin liitetietoihin. Sivulla on painike tallenna ja siirry seuraavaan tositteeseen.

Lopputilanne: Käyttäjän painettua painiketta liitetieto tallennetaan ja vanha poistetaan ensin. Käyttäjälle näytetään seuraava liitetieto, kuten kuvauksen alussa.

9 Pää ja päiväkirjan tarkastelu

Alkutilanne: Käyttäjä on kirjautunut ja painaa linkkiä pää- tai päiväkirja.

Kuvaus: Päiväkirjalinkki ohjaa sivulle, jolla näytetään päiväkirja. Erottavien tositteiden perässä on erottaa linkki, joka vie tositteen muokkaukseen. Pääkirjalinkki ohjaa sivulle, jossa näytetään pääkirja koko tilikaudelta. Sivulla käyttäjä voi valita välin, jolta pääkirja näytetään esimerkiksi arvonlisäveroilmoitusta varten.

Lopputilanne: Käyttäjä näkee haluamansa tiedon. Ei muutoksia tietokantaan.

10 Tilinpäätöksen tarkastelu

Alkutilanne: Käyttäjä on kirjautunut ja painaa linkkiä tilinpäätös.

Kuvaus: Linkki ohjaa sivulle, jossa käyttäjä voi painaa sen vuoden valintanapista, minkä vuoden tilinpäätöksen hän haluaa tulostaa. Hän voi valita toisesta valintanapista vertailuvuoden. Molempiin voidaan syöttää erikseen tilikauden alku- ja loppupäivät. Sivulle tultaessa kentissä on käyttäjän aiemmin antamat arvot. Valintojen jälkeen käyttäjä painaa painiketta tee tilinpäätös. Ennen valintoja käyttäjälle näytetään kulloinkin käytössä oleva tilikausi. Tilikausi, jonka kohdalla lukee nyt, sisältää käytössä olevan tilikauden ja sen käyttöönoton jälkeen käyttäjän tilikauteen tekemät merkinnät.

Lopputilanne: Käyttäjä näkee taseen, tuloslaskelman ja liitetiedot valitsemaltaan vuodelta ja taseen ja tuloslaskelman vertailutiedot valitsemaltaan vuodelta. Ei muutoksia tietokantaan.

11 Tilikohtaisen tuloslaskelman ja taseen tarkastelu

Alkutilanne: Käyttäjä on kirjautunut ja painaa jompaakumpaa linkeistä tilikohtainen tase tai tilikohtainen tuloslaskelma.

Kuvaus: Valittu linkki ohjaa sivulle, joka toimii kuten edellä, mutta vertailuvuotta ei voi valita.

Lopputilanne: Käyttäjä näkee valintansa mukaan joko tilikohtaisen taseen tai tuloslaskelman. Ei muutoksia tietokantaan.

12 Tase-erittelyiden lataaminen omalle koneelle

Alkutilanne: Käyttäjä on kirjautunut ja painaa tase-erittelyt linkkiä

Kuvaus: Valittu linkki ohjaa sivulle, jossa voi valintanapista painamalla valita joko pysyvien tai vaihtuvien vastaavien erät tai oman pääoman tai vieraan pääoman erät tai pakolliset varaukset tai tilinpäätössiirtojen kertymän. Eri tilikausille on omat valintanapit. Painamalla painiketta valintojen jälkeen käyttäjälle näytetään käyttöjärjestelmän latausdialogi, jonka avulla tiedosto ladataan.

Lopputilanne: Käyttäjällä on koneella valitsemansa tiedot csv-tiedostona tase-erittelyiden muokkaamista varten. Ei muutoksia tietokantaan.

13 Varmuuskopioiden lataaminen omalle koneelle

Alkutilanne: Käyttäjä on kirjautunut ja painaa linkkiä varmuuskopioi.

Kuvaus: Valittu linkki ohjaa sivulle, jossa käyttäjä voi valintanapista painamalla valita joko tositetaulun, kirjaustaulun tai omat tilinsä. Painikkeen painaminen käynnistää käyttöjärjestelmän oman latausdialogin.

Lopputilanne: Käyttäjällä on koneella valitsemansa taulu csv-tiedostona. Ei muutoksia tietokantaan.

14 Tililuettelon ja tilikartan tarkastelu

Alkutilanne: Käyttäjä on kirjautunut ja painaa tililuettelo- tai tilikarttalinkkiä.

Kuvaus: Tilikarttalinkki ohjaa sivulle, jossa käyttäjälle näytetään kaikkien käytävissä olevien kirjanpitoilien tilinumerot ja nimet sekä tieto siitä, minkä tilinpäätöstilin alle kukin kirjanpitoili kuuluu. Tililuettelolinkki ohjaa sivulle, jossa näytetään vastaavat tiedot niistä tileistä, joille on tehty kirjauksia. Sivulla on valintaruutu, jonka valitsemalla ja saldovalinnan näppäintä painamalla sivulla näytetään kirjanpitoilien ja tilinpäätöstilien saldot.

Lopputilanne: Käyttäjällä on näytöllään halutut tiedot. Ei muutoksia tietokantaan.

15 Uuden tilikauden avaus

Alkutilanne: Käyttäjä on kirjautunut ja painaa avaa tilikausi -linkkiä. Käytössä on tilikausi numero nolla. Tilikautta yksi ei ole vielä olemassa vaan pelkästään tilikausi nolla tai tilikausi yksi on suljettu ensin eli siirretty tilikaudesta yksi tilikaudeksi kaksi.

Kuvaus: Käyttäjälle näytetään olemassa olevat tilikaudet alkamis- ja päättymispäivineen sekä käytössä olevan tilikauden numero. Käyttäjä antaa kenttiin uuden tilikauden alkamis- ja päättymispäivät sekä painaa uuden tilikauden avauksen painiketta.

Lopputilanne: Tilikauden nolla taulut on kopioitu tilikauden yksi tauluiksi. Sitä ennen nykyiset muutokset on kopioitu tilikauteen nolla. Käyttöön jää tilikausi nolla ja edellisen tilikauden perusteella on tehty avausvienti, jossa tilikauden voittoa ei ole viety edellisten tilikausien voittoon. Käyttäjän tulee huolehtia korjauksesta.

16 Tilikauden sulkeminen

Alkutilanne: Käyttäjä on kirjautunut ja painaa sulje tilikausi -linkkiä. Käyttäjällä on käytössä tilikausi yksi.

Kuvaus: Käyttäjälle näytetään olemassa olevat tilikaudet alkamis- ja päättymispäivineen sekä käytössä olevan tilikauden numero. Käyttäjä painaa tilikauden sulkemisen painiketta.

Lopputilanne: Tilikausien taulut on kopioitu eteenpäin toistensa päälle, uudempi tilikausi vanhemman päälle vanhimmasta tilikaudesta alkaen uudempiin edeten. Jos maksimitaulumäärää ei ole vielä ylitetty, vanhimman tilikauden taulut kopioidaan ensin vieläkin vanhemmiksi. Operaatio suoritetaan tilikauteen yksi asti. Ennen tilikauden yksi kopiointia tehdyt muutokset kopioidaan tilikauteen yksi. Lopuksi tilikauden yksi taulut poistetaan. Käyttöön jää vanha tilikausi eli operaation jälkeen suljettu tilikausi kaksi.

17 Tilikauden vaihtaminen

Alkutilanne: Käyttäjä on kirjautunut ja painaa vaihda tilikausi -linkkiä. Käyttäjällä on käytössä monta tilikautta.

Kuvaus: Käyttäjälle näytetään olemassa olevat tilikaudet alkamis- ja päättymispäivineen sekä käytössä oleva tilikausi. Käyttäjä syöttää sen tilikauden numeron, johon halutaan siirtyä ja painaa painiketta.

Lopputilanne: Halutun tilikauden tiedot kopioidaan käsiteltäviin tauluihin. Käsiteltävään tilikauteen tehdyt muutokset kopioidaan kuitenkin sitä ennen nykyiseen vuoteen, mikäli on kyseessä tilikaudet nolla tai yksi, joita ei siis ole lukittu. Käyttäjällä on käytössä olevissa tauluissa haluamansa tilikauden tiedot.

18 Tilikauden muokkaus

Alkutilanne: Käyttäjä on kirjautunut ja painaa tilikauden muokkaus -linkkiä.

Kuvaus: Käyttäjälle näytetään tilikausien alkamis- ja päättymispäivät. Tilikauden tietoja voi muuttaa, jos sitä ei ole suljettu. Muutosten jälkeen käyttäjä painaa sivulla olevaa painiketta.

Lopputilanne: Alkamisajat ja päättymisajat päivitetään tietokantaan.

Liite 2 Kirjanpito-ohjelman luokkakaavio

